Opinion Diversity Maximization in Social Networks

Sijing Tu

School of Science

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 21.1.2019

Supervisor and advisor

Prof. Aristides Gionis

Aalto University
School of Science

Aalto University
School of Science

Aalto University, P.O. BOX 11000, 00076 AALTO
www.aalto.fi
Abstract of the master's thesis

Author Sijing Tu

Title Opinion Diversity Maximization in Social Networks

Degree programme Computer Science and Communications Engineering

Major Machine Learning and Data Mining **Code of major** SCI3044

Supervisor and advisor Prof. Aristides Gionis

Date 21.1.2019 **Number of pages** 47+1 **Language** English

Abstract

In recent years, the social networks play an important role as the information sources for many people. The algorithms applied by these platforms feed the users with opinions that meet their tastes. These algorithms, on the other hand, harm the free flow of information by forming filter bubbles and echo chambers. The motivation of the thesis is to break filter bubbles by maximizing the diversity of opinion exposures in a social network.

To achieve the goal, we assume that we can change a limited number of users' exposures to the information in the network. We propose two concrete models, bounded-box diversity maximization and the ℓ_2-bounded diversity maximization respectively. Both of them are convex maximization problems subject to different constraints. Besides the common cardinality constraint and linear constraint, the second model has an ℓ_2 constraint. We give a detailed exposition of convex maximization problems under different constraints.

We prove that first problem can be discretized and transformed into the Quadratic Knapsack Problem (QKP), while the second one can not. Guided by convexity and QKP, Greedy Algorithms and Semidefinite Relaxation are applied. In a high level, the core idea of the Semidefinite Relaxation consist of two components: first, relaxing the original problem into Semidefinite Programming and obtaining a solution; second, rounding the solution to a feasible solution of the original problem through sampling from a Gaussian distribution.

To evaluate our problem, we implement the algorithms on datasets that have clear chambers and low diversity structure. We show that semidefinite relaxation based algorithms work well when there are large changes on the cardinality, while when the changes are small, the greedy algorithms are comparable. The greedy algorithms have descent scalability for large datasets.

Keywords filter bubble, diversity maximization, semidefinite relaxation, greedy algorithms, convex maximization

Preface

The journey of finishing the thesis has been long and challenging. Without the help from my supervior, my friends and my family, I could not have accomplished.

First and foremost I would like to express my gratitude to Professor Aristides Gionis for providing me with an opportunity to work in the Data Mining Group, for the valuable suggestions and for the guidance and constant support. Besides, I'd like to thank my partner Antonis Matakos for providing me with this interesting topic and all the discussions we have. I'd like to thank all the friends in the Data Mining Group for the general company and discussions.

Then I'd like to express my thanks to my friends. Jiayao, Weiqing, Lan, Siyi, who companied me, helped me for all those days in Finland. Fu, Tommi, Dimitrij and all the friends who weekly played table tennis and badmintons with me.

Last but not least, I'd like to express my gratitude to Shanshan and my parents for always trusting me, supporting me and cheering me forward during the hardest days.

Otaniemi, 21.01.2019

Sijing Tu

Contents

Symbols, Operators and Abbreviations

Symbols

$\mathbf{s}$	Exposure Vector
η	Diversity Index
k	Cardinality Constraint
α	ℓ_2 Constraint

Operators

$\mathbf{x}$	Vector $\mathbf{x}$
$\mathbf{X}$	Matrix $\mathbf{X}$
$\langle \mathbf{X}, \mathbf{Y} \rangle$	The Frobenius inner product of $\mathbf{X}$ and $\mathbf{Y}$
$diag(\mathbf{X})$	The diagonal vector of matrix $\mathbf{X}$
$Diag(\mathbf{x})$	The diagonal matrix whose diagonal equals to $\mathbf{x}$
$Diag(\mathbf{X})$	The diagonal matrix whose diagonal equals to $diag(\mathbf{X})$
$rank(\mathbf{A})$	Rank of a matrix $\mathbf{A}$
$card(\mathbf{x})$	Number of non-zeros elements in a vector $\mathbf{x}$
$\mathbf{e}$	A vector of all ones

Abbreviations

SDP	Semidefinite Programming
QCQP	Quadratically Constrained Quadratic Programming
IQP	Integer Quadratic Programming

1 Introduction

In the contemporary society, the personalized recommendation systems influence our lives in almost every aspect. Whenever we listen to music, watch TV series or play video games, the platform always advertise several items we might be interested in. Similarly, the social networks push us with content and the opinion leaders that meet our tastes based on our previous behaviors on the platform. While the recommendation of the leisure stuff makes our life easier and enjoyable, the recommendation of the opinions are not usually beneficial, from the perspective of society.

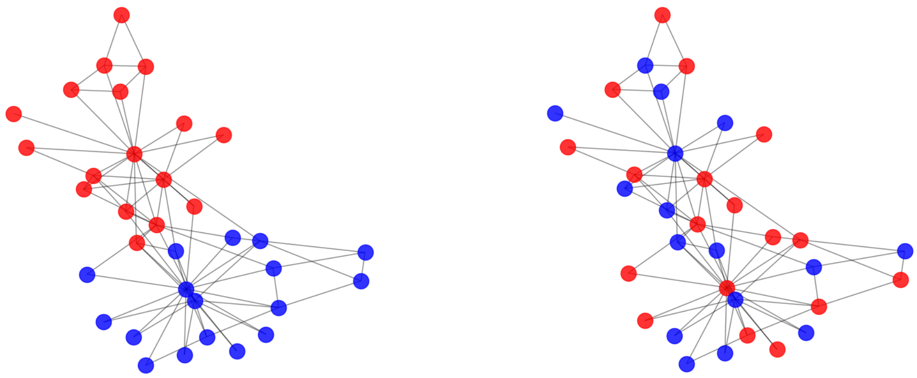
According to a research [1] conducted by the Pew Research Center in 2018, more than 60% of the American adults occasionally get news from the social media; among which 30% often do so. On the other hand, the drawbacks of the social networks are being revealed through the phenomenon *political polarization*. Google memo [2] has drawn our attention how our opinions diverse from each other among the social issues. Another research [3] conducted by Pew Research Center showed that the ideology shared by the American public has gone into extremes; which results in the irreconcilable contradictions on social issues, such as gun control, the same-sex marriage, and the immigration policies.

*Political polarization*s repeatedly happened in the history. Taking the problem extremely, the so called Social Conflict by Karl Marx described irreconcilable polarization among the workers and the capitalists in Capitalism, and inspired revolutions in various countries. In the current society, the *social networks* have become one of the majority platforms of obersing *political polarization*. Though they publicize themselves as the platform of free flow of the information, the flow is harmed by the current algorithms applied in the social network platforms that emphasis their users' *confirmation bias* [4]. It keeps recommending the content that users already believe in, and avoid challenging their beliefs. This intellectual isolation resulted in the constantly tailed content recommendations based on the users' tastes is called *filter bubble*s [5]. The like-minded people in the social networks form *echo chamber*s [6] in which the similar opinions are amplified and refined, and as a consequence, the *fake news* spread. As shown in the report [6] conducted in 2015, the ratio of seeing cross-cutting content to the ideologically consistent content is less than 10% among both the conservatives and the liberals.

The blocked communications on different opinions accelerates such phenomena. The domain specific empirical researches [7, 8, 9] in different social networks have been conducted. We can roughly characterize the more general researches on *political polarization* into three aspects: 1. The formation of the polarization [10]; 2. The measure of polarization [11, 12, 13]; 3. The solutions, which include methods connecting different opinions in the social network [14], minimizing the polarization index [12, 15], and diversifying different information exposure [16, 17, 18].

1.1 Motivation

In this thesis, we are motivated to mitigate the *filter bubble* and break the *echo chambers* in the social network. To do so, we change some of the current information that users are exposed to such that the users and the their friends are exposed to different opinions. To make the problem intuitive, we use Zachary's karate club to give a simple example. Let the color red and blue represent different opinions regarding an issue in the social network. In the left side Figure 1a, the same opinion is spreading inside each group while blocked between the groups, there are two echo chambers formed in the network. While in Figure 1b, the opinions are more equally distributed.



(a) Two echo chambers network

(b) No echo chambers

Figure 1a resembles the current situation in the social network that the users are *isolated* in their *filter bubbles* and form *echo chambers*. While Figure 1b is the ideal situation we want to achieve: users have lots of connections with friends who are exposed to different opinions.

1.2 Research Scope

The thesis is an extension of the previous work of Matakos *et al.* [18]. They asked the following question in their work.

> Given a fixed amount of content-recommendation activity, which users should we target and completely change the content they receive, so as to maximally increase diversity?

In their work, they assume that the opinion exposure of each user is quantifiable and is of binary from: either -1 or 1. In our work, we cast the value into an range $[-1, 1]$. We proposed two models to solve the following question.

> Given a fixed amount of content-recommendation activity, which users should we target and how much change on the content should we make, so as to maximally increase diversity?

To make our model reasonable, we make the following assumptions:

- We consider the mutual friendship connections in the social network, we assume that the social network can be simplified as a weighted simple undirected graph, where the edge refers the connection, the node refers to the user, and the weight refers to the closeness index;
- We assume that the opinion exposure of each user is quantifiable and can be cast into the area $[-1, 1]$, where the -1 and 1 represent the two extreme cases respectively.
- We assume that our *recommendation system* can change the users' exposure to the information directly, the operability is out of the scope in the master thesis.
- We assume that the friends will influence the oponions of each other in the social network.

1.3 Contribution

- We introduce two problems that extend Matakos *et al.*'s work [18] into the continuous exposure space. We show that in the first variant the optimal solution is reached when the new exposure vector takes extreme values, which resembles the discrete problem. We also propose a second variant with an added $\ell 2$-constraint, which prevents the exposure vector from taking extreme values.
- We extend our empirical evaluation to incorporate the two new problem formulations. We report the achieved increase in the diversity index from the two new problem formulations. Additionally, we demonstrate the cost of allowing the exposure index to take values in the continuous space, as well as from imposing $\ell 2$ and Box constraints.

1.4 Structure of the Thesis

- In Chapter 2, we introduce the original work and conduct the literature review in the related areas;
- In Chapter 3, we conduct a through survey on the related methods that are relevant with our thesis;
- In Chapter 4 and Chapter 5, we formulate and propose methods to solve the Problem 4.1, and Problem 5.1 respectively;

- In Chapter 6, we conduct the experiments to verify our methods;
- In Chapter 7, we analyze the drawbacks of our model and propose the future work.

1.5 Acknowledgement

The work is based on the paper of Matakos *et al.*'s work [18], lot of methods discussed in the thesis are inspired by their work.

2 Background

2.1 Maximize Opinion Diversity in Social Network

In this section, we briefly discuss the work of Matakos *et al.* [18] on how the problem is formulated and the open problems in the paper. As we discussed in the Chapter 1, the recommendation system works as a driver of polarization since it keeps feeding its users with information that they believe in, strengthens their confirmation biases and forms the politically homogeneous groups as a result. Assuming that the overall opinion tendency each user exposed to is quantifiable and has a binary form, the paper seeks to avoid the formation of homogeneous groups by changing a limited number of users' exposures, such that the users exposed to different opinions have as many connections with each other as possible. This work is different from the work of Musco *et al.* [15], the latter one changes the network structure to avoid disputes while the work of Matakos *et al.* [18] keeps the network structure, changes the attributes and maximize the diversity.

2.1.1 Problem Formulation

To make the model feasible, the complex social network is simplified as a simple undirected weighted graph $G = (V, E, w)$, where the w is a function that maps edges in E to possitive values. Let $\mathbf{A}$ denote the adjacency matrix of G.

The thesis defines a binary formed *exposure index* to measure overall opinion tendency that a user exposed to in the social network, namely, the user's exposure can be either -1 or 1. For instance, according to a research [19] conducted by the Pew Research Center, 47% of the consistent conservatives cite Fox News (a source of conservative views) as their main source for news about politics; while the consistently liberal regard NPR, PBS, BBC (the sources of liberal views) as their trusted news. The -1 and 1 respectively correspond to these two extreme cases. Denote $\mathbf{s} \in \{-1, 1\}^n$ to be a vector of the *exposure index* of all the users in the social network.

The *diversity index* is defined to measure the overall connections between the opposite opinion exposures in the social network. The *diversity index* equals to the pairwise squared differences between the connected *exposure index*es, the edge weight is counted in as well, the larger the weight, the more influence it has on the *diversity index*. Formally, the definition is given in Definition 2.1.

Definition 2.1 (Diversity Index)**.** *Given a graph G and its exposure index vector $\mathbf{s}$, the diversity index of the graph G is written as $\eta(G, \mathbf{s})$, where* $\eta(G, \mathbf{s}) = \sum_{(i,j) \in E} A_{ij}(\mathbf{s}_i - \mathbf{s}_j)^2$.

The main task of the paper is to maximize this *diversity index* in a given graph by changing a given number of *exposure index*es, as in Problem (2.1).

Problem 2.1 (Diversity Maximization [18])**.** *Given a graph $G = (V, E, w)$ and an exposure index vector $\mathbf{s} \in \{-1, 1\}^n$, change at most k elements in $\mathbf{s}$ to get a new vector $\mathbf{y}$, such that the new diversity index $\eta(G, \mathbf{y})$ is maximized.*

The Problem (2.1) has a direct mathematical programming form as in Equation (2.1), where *card* denotes the number of non-zero elements in the vector.

$$\begin{aligned} \underset{\mathbf{y}}{\text{maximize}} \quad & \eta(G, \mathbf{y}) \\ \text{subject to} \quad & \mathbf{y} \in \{-1, 1\}^n \\ & card(\mathbf{y} - \mathbf{s}) \leq k \end{aligned} \tag{2.1}$$

2.1.2 Solutions and Open Problems

The Problem (2.1) can be transformed into a Quadratic Knapsack Problem (QKP) and the latter is proved to be a strongly NP-Hard problem [20]. A semidefinite relaxation as in [21] is applied to solve the problem; to draw the feasible results, the rounding technique in [22] is applied.

The open problem is that assuming that the *exposure index* is continuous, and initiating the entries in $\mathbf{s}$ to be between -1 and 1, how can we build the model to solve the problem? The thesis focuses solving such extension.

2.2 Related Work on Polarization

There are plenty of papers about polarization-related issues in the recent years. The thesis describes these papers in three aspects. The formation of polarization, measuring the polarization, and methods to mitigate polarization.

2.2.1 Formation

The polarization models have their bases on the *opinion formation* models. As early as the 1970s, DeGroot's seminal work [23] on reaching a consensus among a group of people built a stochastic process model on a graph to simulate the process, each node on the graph is assigned by a prior opinion with the form of probability at the beginning, and is repeatedly renewed by the weighted average on all the opinions. As an extension of DeGroot's work that completely changes the opinions of nodes at each iteration, Friedkin *et al.* [24] assumes a inner belief of each node that always keeps as a constant during the process. More recently, Dandekar *et al.* [10] proved that extreme opinioins do not form in the DeGroot's opinion averaging model and adapts the model to simulate the polarization in a group.

2.2.2 Measuring Polarization

There are numerous papers [7, 8, 9, 11, 12, 25, 26] measuring polarization on different sceneries. Community detection method, such as modularity, has been directly applied in detecting the polarization of parties in congress [25], and of left and right leaning users on Twitter [7]. However, it is not an ideal tool to detect the polarization in general. Firstly, the community partitions do not mean the polarization; secondly, it measures the connections inside the communities as well. To solve the latter problem, Guerra *et al.* [11] proposes a measure that is only based on the differences between the community boundaries. Classfied according to different situations, there are quantifiable analysis on consumption patterns of the same video on different platforms [9], the citation behaviors of the liberal and conservative blogs [8]. From the perspective of methods, there is a paper that formulates the evaluation of political polarization as a node classification and ranking problem on signed graph [26]; and one [12] extends the opinion fromation model and defines a *polarization index* in their paper. Recently, Garimella *et al.* [13] proposes a systematic and general method to quantify the controversy. In their paper, they give a pipeline to build the graph, partition the graph, and measure the controversy. As one of the results, they find out that the graphical structure based features outperforms the content based feature in capturing the controversy.

2.2.3 Mitigating Polarization

To reduce polarization, there are different settings in the recent papers. For instance, the idea of building the models from the perspective of maximizing the information exposure [16, 18, 17]. Garimella *et al.* [16] assumes a situation that two opposed opinions are spreading the network, and it tries to balance the information exposure in the network, Matakos *et al.* [18] discusses how to maximize the opinion diversity of the social network by changing a budget of the information exposure of its users. This master thesis work is based on the work of Matakos *et al.* [18]. Aslay *et al.* [17] models to maximize the information exposure as well while the setting is based on the influence propagation. The paper assumes that the users in the social network can be influenced by their neighbors with some probability, and the model is formulated in the context of information propagation. The theory of the paper can be separated into two parts. In the first part they prove that the programming is to maximize a submodular function subject to matroid constraints, and the proposed greedy algorithm achieves an approximation guarantee of $\frac{1}{2}$ [27]. In the second part they propose a scalable algorithm that can compute the expected influence spread as the subroutine of the greedy algorithm.

Different than these three models, Musco *et al.* [15] discusses the structure of a network that can minimize the polarization and disagreements. In that work, they propose two problems, respectively, how to change the weights and how to change the opinions with a given limit to minimize the disagreement. It is worth mentioning that the *Network Disagreement Index* in Musco *et al.*'s work has the same formulation as *Diversity Index* defined in Matakos *et al.*'s work [18].

3 Research Methods

In this section we discuss the technical details that are relevant to this master thesis.

3.1 Convex Optimization

In order to solve the problems we propose in the thesis, we relax the original problems into *Semidefinite Programming* (SDP) problems. The latter one belongs to the family of the *convex optimization* problems. This section introduces some basic facts about these two concepts. To illustrate the *Convex Optimization* problem, we start by defining the convex sets and convex functions.

Definition 3.1 (Convex Set [33])**.** *A set $\mathbb{S}$ in $\mathbb{R}^n$ is said to be convex if the line segment joining any two points of the set also belongs to the set.*

Definition 3.2 (Convex Function [33])**.** *Let $f : \mathbb{S} \to \mathbb{R}$, where $\mathbb{S}$ is a nonempty convex set in $\mathbb{R}^n$. The function f is said to be convex on $\mathbb{S}$ if for any $1 \geq \lambda \geq 0$:*

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda)f(x_2)$$

We use the notation in Boyd's *et al.*'s book [28, Ch.4, p.127], a convex optimization problem can be written as the following Equation (Convex Optimization).

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m, \\ & h_i(x) = 0, \quad i = 1, \ldots, p, \end{array} \tag{Convex Optimization}$$

where f_0 and $f_i, i = 1, \ldots, m$ are convex functions, and $h_i, i = 1, \ldots, p$ are affine functions of the form $h_i(x) = a_i x - b_i$. The feasible area is convex. Generally speaking, convex optimization problem can be regarded as minimizing a convex function over a convex set.

3.1.1 The Strength of Convex Optimization

For the past decades, many real world problems have been modeled as convex optimization problems, for instance, the Markowitz portfolio optimization problem can be modeled as a quadratic minimization problem given an affine equality constraint. Advantages of Convex Optimization problems include special geometry structure, its local optimal solution is also its global optimal solution. For the non-constrained problems, even the basic gradient descent method has a good convergence behavior. For the constrained problems, the well developed prime-dual based interior-point method can be used to solve the problems of small data size, and method of Alternating Direction Method of Multipliers (ADMM) can be applied to solve the larger ones.

3.1.2 Semidefinite Programming

Semidefinite programming (SDP) is a special case of convex optimization. We use the notation given by Vandenberghe *et al.* [29] to describe SDP problems. For any matrix $\mathbf{X}$, let $\mathbf{X} \succcurlyeq 0$ denote $\mathbf{X}$ a positive semidefinite matrix, which means for all $\mathbf{v} \in \mathbb{R}^n$, $\mathbf{v}^\top \mathbf{X} \mathbf{v} \geq 0$. As in Equation (SDP),

$$\begin{aligned} &\text{minimize} && \mathbf{c}^\top \mathbf{x} \\ &\text{subject to} && \mathbf{F}(\mathbf{x}) \succcurlyeq 0, \end{aligned} \tag{SDP}$$

where $\mathbf{F}(\mathbf{x}) = \mathbf{F}_0 + \sum_{i=1}^{m} x_i \mathbf{F}_i$. $\mathbf{F}_i \in \mathbf{R}^{n \times n}, i = 1, \ldots, m$ are symmetric matrices, and the $\mathbf{c} \in \mathbf{R}^m$ is a known vector. The feasible area of the programming is convex, since for any $\mathbf{x}_1$ and $\mathbf{x}_2$ with $\mathbf{F}(\mathbf{x}_1) \succcurlyeq 0$ and $\mathbf{F}(\mathbf{x}_2) \succcurlyeq 0$, given a λ with $1 \geq \lambda \geq 0$, $\mathbf{F}(\lambda \mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2) \succcurlyeq 0$; the objective is to minimize an affine function, thus the programming is convex programming.

The semidefinite programming is a general form of linear programming, since the condition $\mathbf{F}(\mathbf{x}) \succcurlyeq 0$ is infinite combinations of the linear constraints, namely for all $\mathbf{v} \in \mathbb{R}^n, \langle \mathbf{F}(\mathbf{x}), \mathbf{v}\mathbf{v}^\top \rangle \geq 0$. Like linear programming, once the problem has been formulated as semidefinite programming, it takes polynomial time to solve the problem using the dual-prime interior-point method. There are also surveys [29, 30] of these interior-point methods.

The Dual Problem (SDP-Dual) is a SDP as well. Semidefinite programming can be applied to solve NP-Hard problems in combinatorial optimization areas, to name a few, the general 0-1 integer programs [31], the graph cut problems [32] and Quadratic knapsack problems [21].

$$\begin{aligned} &\underset{\mathbf{Z}}{\text{maximize}} && -\langle \mathbf{F}_0, \mathbf{Z} \rangle \\ &\text{subject to} && \langle \mathbf{F}_i, \mathbf{Z} \rangle = c_i, i = 1, 2, \ldots, m, \\ &&& \mathbf{Z} \succcurlyeq 0. \end{aligned} \tag{SDP-Dual}$$

In this thesis, we are especially interested in the quadratic knapsack problem, as we will discuss in Section 3.4.2.

3.2 Convex Maximization

Let f be a convex function, let $\mathbb{K}$ be a convex set, let $\mathbf{x}$ be a vector variable, the convex maximization problem can be written as Equation (Convex-max).

$$\begin{aligned} &\underset{\mathbf{x}}{\text{maximize}} && f(\mathbf{x}) \\ &\text{subject to} && \mathbf{x} \in \mathbb{K}. \end{aligned} \tag{Convex-max}$$

The research on convex maximization problems are not as extensive as on convex optimization problems, it seems the maximization problems do not have sufficient geometry information to apply and are regarded in general as harder problems. Nevertheless, we have a clue to solve the problem when the feasible set is a compact polyhedral set, since the optimal variables are always at the vertices of the polyhedron. These *vertices* are called *extreme points* of the polyhedral set. The original programming is reduced to searching these extreme points, which are very similar with the simplex methods that are applied in linear programming.

Let $\mathbf{P}$ be an $m \times n$ matrix, let α be an $m \times 1$ vector, the polyhedral set $\mathbb{S} = \{\mathbf{x} : \mathbf{Px} \leq \alpha\}$ is the intersection of the hyperspaces. Let $\mathbf{x}$ be an extreme point of $\mathbb{S}$, then for all $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{S}$, if $\mathbf{x} = \lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2, 0 \leq \lambda \leq 1$, either $\lambda = 1$ or $\lambda = 0$ or $\mathbf{x} = \mathbf{x}_1 = \mathbf{x}_2$.

According to the Representation theorem [33, Ch.2, p.72], if the polyhedral set is bounded, any point in the polyhedral set can be written as a convex combination of its extreme points. Let $\mathbb{S}$ be a polyhedral set in $\mathbb{R}^n$, let $\mathbf{x}$ be any point in $\mathbb{S}$, then $\mathbf{x} = \sum_{i=1}^{n+1} \lambda_i \mathbf{x}_i$, where $\mathbf{x}_i, i = 1, \dots, n+1$ are extreme points and $\lambda_i, i = 1, \dots, n+1$ are non-negative values where $\sum_{i=1}^{n+1} \lambda_i = 1$. In Theorem 3.1, we formally show that the optimal solutions are among the extreme points of the polyhedral set.

Theorem 3.1 (Bazaraa *et al.* [33])**.** *Let $f : \mathbb{R}^n \to R$ be a convex function, and let $\mathbb{S}$ be a nonempty compact polyhedral set in $\mathbb{R}^n$. Consider the problem to maximize $f(\mathbf{x})$ subject to $\mathbf{x} \in \mathbb{S}$. An optimal solution $\overline{\mathbf{x}}$ to the problem then exists, where $\overline{\mathbf{x}}$ is an extreme point of $\mathbb{S}$.*

Proof. Let $\mathbf{x}$ be any point in the convex set, let $\mathbf{x}_1, \dots, \mathbf{x}_{n+1}$ be $n+1$ extreme points such that $\mathbf{x}$ can be written as a convex combination of $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{n+1}$. $\mathbf{x} = \sum_{i=1}^{n+1} \lambda_i \mathbf{x}_i$ such that $\sum_{i=1}^{n+1} \lambda_i = 1$.

$f(\mathbf{x}) = f(\sum_{i=1}^{n+1} \lambda_i \mathbf{x}_i) \leq \sum_{i=1}^{n+1} \lambda_i f(\mathbf{x}_i) \leq \sum_{i=1}^{n+1} \lambda_i \max\{f(\mathbf{x}_i) : i = 1, \dots, n+1\} = \max\{f(\mathbf{x}_i) : i = 1, \dots, n+1\}$. □

We give some examples of the convex maximization problems with certain constraints.

3.2.1 Box Constraint

Denote by $\mathbf{c}_1, \mathbf{c}_2 \in \mathbb{R}^n$ two vectors, and $\mathbf{x} \in \mathbb{R}^n$ a vector variable. Let $\mathbf{c}_1 \leq \mathbf{x} \leq \mathbf{c}_2$ be a box constraint. Denote $\mathbb{S}$ as the set $\{\mathbf{x} \mid \mathbf{c}_1 \leq \mathbf{x} \leq \mathbf{c}_2\}$, then $\mathbb{S}$ is a polyhedron set, since it can be equivalently formulated as $S = \{\mathbf{x} \mid \mathbf{Px} \leq [-\mathbf{c}_1^\top, \mathbf{c}_2^\top]^\top\}$, where $\mathbf{P}$ is a $2n \times n$ matrix and

$$P_{i,j} = \begin{cases} -1 & \text{for} \quad 1 \leq i \leq n, j = i, \\ 1 & \text{for} \quad n+1 \leq i \leq 2n, j = i-n, \\ 0 & \text{for} \quad \textit{otherwise}. \end{cases}$$

Let $\mathbf{c}$ be the extreme points of $\mathbb{S}$, then $c_i = c_{1i}$, or $c_i = c_{2i}$, for $i = 1, \dots, n$. There are in total 2^n possibilities of $\mathbf{c}$. In the convex maximization scenario, usually the cutting-plane method will be applied to quickly reduce the solution area, and find the result.

3.2.2 Quadratic Constraint

Quadratic constrained convex maximization problems are harder than box constrained problems. The feasible area is no longer a polyhedron, and as a result, the problem can not be solved by traversing the extreme points.

In Problem (3.1), we formulate a quadratically constrained quadratic maximization problem, where $\mathbf{L}$ is a positive semidefinite $n \times n$ matrix, $\mathbf{s}$ is an $n \times 1$ vector, α is a positive value, and $\mathbf{x}$ is an $n \times 1$ vector variable. Consistently with the problem we are going to solve in the thesis, we consider a simple form of the constraint as $\mathbf{x}^\top \mathbf{x} \leq \alpha$.

$$\begin{aligned} \underset{\mathbf{x}}{\text{maximize}} \quad & \mathbf{x}^\top \mathbf{L}\mathbf{x} + 2\mathbf{s}^\top \mathbf{L}\mathbf{x} + \mathbf{s}^\top \mathbf{s} \\ \text{subject to} \quad & \mathbf{x}^\top \mathbf{x} \leq \alpha. \end{aligned} \tag{3.1}$$

One trivial setting is when $\mathbf{x} \in \mathbb{R}^1$, this problem can be immediately solved by letting $\mathbf{x} = -\sqrt{a}$ or $\mathbf{x} = \sqrt{a}$. However, $\mathbf{x}$ is usually a vector of dimension much larger than 1. Another special solvable case is that $\mathbf{s} = \mathbf{0}$, then the problem reduces to Equation (3.2).

$$\begin{aligned} \underset{\mathbf{z}}{\text{maximize}} \quad & \sqrt{\alpha} R(\mathbf{z}, \mathbf{L}) \\ \text{subject to} \quad & R(\mathbf{L}, \mathbf{z}) = \frac{\mathbf{z}^\top \mathbf{L}\mathbf{z}}{\mathbf{z}^\top \mathbf{z}}, \\ & \mathbf{x} = \pm\sqrt{\alpha}\mathbf{z}, \end{aligned} \tag{3.2}$$

where $R(\mathbf{L}, \mathbf{z})$ is the Rayleigh-Ritz ratio [34] of the positive semidefinite matrix $\mathbf{L}$. The problem is not harder than the eigendecomposition problem, and the time complexity is bounded by $\mathcal{O}(n^3)$. However, the problem is more relevant when $\mathbf{s} \neq \mathbf{0}$.

Dual form

We can derive a dual of Equation (3.1) as follows. Let λ be a dual variable, the Lagrangian of the Equation (3.1) is:

$$\begin{aligned} L(\mathbf{x}, \lambda) &= \mathbf{x}^\top \mathbf{L}\mathbf{x} + 2\mathbf{s}^\top \mathbf{L}\mathbf{x} + \mathbf{s}^\top \mathbf{s} + (\alpha - \mathbf{x}^\top \mathbf{x})\lambda \\ &= \mathbf{x}^\top (\mathbf{L} - \lambda \mathbf{I})\mathbf{x} + 2\mathbf{s}^\top \mathbf{L}\mathbf{x} + \mathbf{s}^\top \mathbf{s} + \alpha\lambda. \end{aligned}$$

The dual function is:

$$g(\lambda) = \begin{cases} -\mathbf{s}^\top \mathbf{L}(\mathbf{L} - \mathbf{I}\lambda)^\dagger \mathbf{L}\mathbf{s} + \mathbf{s}^\top \mathbf{s} + \alpha\lambda & \text{for} \quad \mathbf{L} - \mathbf{I}\lambda \prec 0 \\ +\infty & \text{for} \quad \mathbf{L} - \mathbf{I}\lambda \not\preccurlyeq 0 \end{cases}.$$

Thus the dual problem can be written as Equation (3.3).

$$\begin{aligned} \underset{\lambda}{\text{minimize}} \quad & -\mathbf{s}^\top \mathbf{L}(\mathbf{L} - \mathbf{I}\lambda)^\dagger \mathbf{L}\mathbf{s} + \mathbf{s}^\top \mathbf{s} + \alpha\lambda \\ \text{subject to} \quad & \mathbf{L} - \mathbf{I}\lambda \prec 0. \end{aligned} \tag{3.3}$$

Using Schur's complement, this dual form can be formulated equivalently as a Semidefinite Programming (SDP) problem in Equation (3.4).

$$\begin{aligned} \underset{t}{\text{minimize}} \quad & t \\ \text{subject to} \quad & \begin{bmatrix} \mathbf{I}\lambda - \mathbf{L} & \mathbf{L}\mathbf{s} \\ \mathbf{s}^\top \mathbf{L} & \mathbf{s}^\top \mathbf{s} + \alpha\lambda + t \end{bmatrix} \succcurlyeq 0. \end{aligned} \tag{3.4}$$

Boyd *et al.* [28, Appendix B] proves that when QCQP has one quadratic constraint, the strong duality holds for the prime and the dual if the condition $\mathbf{x}^\top \mathbf{x} < \alpha$ is feasible (Slater's condition), more interestingly, the semidefinite relaxed problem as in Problem (3.5) holds the strong duality with the Problem (3.4) and Problem (3.1) as well. Problem (3.5) can be solved in polynomial time.

$$\begin{aligned} \underset{\mathbf{X}}{\text{minimize}} \quad & \langle \mathbf{L}, \mathbf{X} \rangle + 2\mathbf{s}^\top \mathbf{L}\mathbf{x} + \mathbf{s}^\top \mathbf{s} \\ \text{subject to} \quad & \langle \mathbf{X}, \mathbf{I} \rangle - \alpha \leq 0, \\ & \begin{bmatrix} \mathbf{X} & \mathbf{x} \\ \mathbf{x}^\top & 1 \end{bmatrix} \succcurlyeq 0. \end{aligned} \tag{3.5}$$

3.2.3 Cardinality and Quadratic Constraints

Connected with the problem we study in the master thesis, we consider an even more interesting setting that puts the cardinality constraint and the quadratic constraint together. We add the cardinality constraint on $\mathbf{x}$ to Problem (3.1) by letting $card(\mathbf{x}) \leq k$, where k is a real positive value, $card(\mathbf{x})$ denotes the number of non-zero elements in $\mathbf{x}$. The problem is formulated as Problem (3.6).

$$\begin{aligned} \underset{\mathbf{x}}{\text{maximize}} \quad & \mathbf{x}^\top \mathbf{L}\mathbf{x} + 2\mathbf{s}^\top \mathbf{L}\mathbf{x} + \mathbf{s}^\top \mathbf{s} \\ \text{subject to} \quad & ||\mathbf{x}||_2^2 \leq \alpha, \\ & card(\mathbf{x}) \leq k. \end{aligned} \tag{3.6}$$

Lemma 3.2. *Let* $\mathbf{X}$ *be an* $n \times n$ *matrix, let* $\mathbf{e}$ *be an* $n \times 1$ *vector of all ones. Problem* (3.7) *is a relaxed form of Problem* (3.6).

$$\begin{aligned} \underset{\mathbf{x},\mathbf{X}}{\text{maximize}} \quad & \langle \begin{bmatrix} \mathbf{X} & \mathbf{x} \\ \mathbf{x}^\top & 1 \end{bmatrix}, \begin{bmatrix} \mathbf{L} & \mathbf{L}\mathbf{s} \\ \mathbf{s}^\top L & \mathbf{s}^\top \mathbf{L}\mathbf{s} \end{bmatrix} \rangle \\ \text{subject to} \quad & \begin{bmatrix} \mathbf{X} & \mathbf{x} \\ \mathbf{x}^\top & 1 \end{bmatrix} \succcurlyeq 0, \\ & \mathbf{e}^\top |\mathbf{X}| \mathbf{e} \leq k\alpha, \\ & \langle \mathbf{X}, \mathbf{I} \rangle \leq \alpha. \end{aligned} \tag{3.7}$$

Proof. By letting $\mathbf{X} = \mathbf{x}\mathbf{x}^\top$, or equivalently put it as $rank(\begin{bmatrix} \mathbf{X} & \mathbf{x} \\ \mathbf{x}^\top & 1 \end{bmatrix}) = 1$. We first write an equivalent form in Problem (3.8).

$$\begin{aligned} \underset{\mathbf{x},\mathbf{X}}{\text{maximize}} \quad & \langle \begin{bmatrix} \mathbf{X} & \mathbf{x} \\ \mathbf{x}^\top & 1 \end{bmatrix}, \begin{bmatrix} \mathbf{L} & \mathbf{Ls} \\ \mathbf{s}^\top\mathbf{L} & \mathbf{s}^\top\mathbf{Ls} \end{bmatrix} \rangle \\ \text{subject to} \quad & rank(\begin{bmatrix} \mathbf{X} & \mathbf{x} \\ \mathbf{x}^\top & 1 \end{bmatrix}) = 1, \\ & \langle \mathbf{X}, \mathbf{I} \rangle \leq \alpha, \\ & card(\mathbf{x}) \leq k. \end{aligned} \tag{3.8}$$

The rank condition can be relaxed into the condition $\begin{bmatrix} \mathbf{X} & \mathbf{x} \\ \mathbf{x}^\top & 1 \end{bmatrix} \succcurlyeq 0$. Besides, D'aspremont *et al.* [35] uses the Cauchy–Schwarz inequality to relax the cardinality constraint. Since $||\mathbf{x}||_1^2 \leq card(\mathbf{x})||\mathbf{x}||_2^2$, $||\mathbf{x}||_1^2 = \mathbf{e}^\top|X|\mathbf{e}$, $||\mathbf{x}||_2^2 = trace(X)$, and $card(\mathbf{x}) \leq k$, we can relax the $card(\mathbf{x}) \leq k$ into $\mathbf{e}^\top|X|\mathbf{e} \leq k\alpha$. Putting together, we have the relaxed problem of Problem (3.7). □

3.3 Non-Convex Quadratic Programming

Besides being applied in the convex maximization problems, the semidefinite relaxation can be applied to solving the general non-convex quadratic programming, for instance, the programming with box constraint.

3.3.1 Box Constraints

Denoting by $\mathbf{P}$ any real matrix, and by $\mathbf{x}$ a vector variable, the box-constrained convex maximization problem can be formulated as Problem (3.9).

$$\begin{aligned} \underset{\mathbf{x}}{\text{maximize}} \quad & \mathbf{x}^\top\mathbf{P}\mathbf{x} \\ \text{subject to} \quad & ||\mathbf{x}||_\infty \leq 1. \end{aligned} \tag{3.9}$$

Ye [36] relaxed the problem into a semidefinite programming problem as in Problem (3.10). After solving the relaxed SDP problem, a randomized algorithm was applied to extract the variable out of the positive semidefinite matrix.

$$\begin{aligned} \underset{\mathbf{X}}{\text{maximize}} \quad & \langle \mathbf{P}, \mathbf{X} \rangle \\ \text{subject to} \quad & ||diag(\mathbf{X})||_\infty \leq 1, \\ & \mathbf{X} \succcurlyeq 0. \end{aligned} \tag{3.10}$$

Denoting by $\mathbf{X}^*$ the optimal solution of Problem (3.10), and by $\mathbf{x}^*$ the optimal solution of Problem (3.9). Similar with the method in the work of Goemans and Williamson [32], Ye [36] applied a randomized procedure to extract an approximated solution $\mathbf{x}^*$ out of $\mathbf{X}^*$. First, it decomposed $\mathbf{X}^*$ into the multiplication of a matrix and its transpose, denote the matrix by $\mathbf{V}$, $\mathbf{X}^* = \mathbf{V}^\top\mathbf{V}$; second, it draw a random vector $\mathbf{u} \in \mathcal{N}(\mathbf{0}, Diag(\mathbf{e}))$; third, $\mathbf{x}^*$ was approximated by $\mathbf{x} = \sqrt{Diag(\mathbf{X}^*)}sign(\mathbf{V}^\top\mathbf{u})$.

Let $\mathbf{x}_{\min}$ be the minimum optimal solution of (3.9), it is proved in the papper that $\dfrac{\mathbf{x}^* - \mathbb{E}[\mathbf{x}]}{\mathbf{x}^* - \mathbf{x}_{\min}} \leq \dfrac{\pi}{2} - 1$.

3.4 Integer Programming

In this section, we give two concrete examples on how the semidefinite programming can be applied to solve the integer programmings, respectively the Max cut and the quadratic knapsack problems.

3.4.1 Max Cut

Let us consider a setting in a simple undirected weighted graph $G = (V, E, w)$. $V = \{1, 2, \ldots, n\}$. Let $\mathbf{A}$ be the adjacency matrix of this weighted graph, let $\mathbf{D}$ be a diagonal matrix, with $D_{i,i} = \sum_j A_{i,j}$.

Problem 3.3. *(Max Cut) Given a simple undirected weighted graph $G = (V, E, w)$, partition V into two subsets V_1 and V_2 such that the summation of the weights of the edges between V_1 and V_2 is maximized.*

Let $\mathbf{x}$ be a vector such that $x_i = 1$ if i is partitioned into V_1, $x_i = -1$ if i is partitioned into V_2. The max cut problem can be formulated as:

$$\begin{aligned} \underset{\mathbf{x}}{\text{maximize}} \quad & \frac{1}{2} \sum_{i<j} A_{i,j}(1 - x_i x_j) \\ \text{subject to} \quad & \mathbf{x} \in \{-1, 1\}^n. \end{aligned} \tag{3.11}$$

Richard M. Karp classified the problem as a NP-Complete problem [37], thus proving the hardness of finding an optimal solution. Goemans and Williamson [32], instead of solving the problem (3.11) directly, employed the result of a polynomial time solvable problem (3.12) to develop a randomized algorithm. The partition is achieved by uniformly randomly picking a vector $\mathbf{u} \in \mathbb{R}^n$, and placing i into V_1 if $\mathbf{u}^\top \mathbf{v}_i \leq 0$, otherwise, i is placed into V_2. This randomized algorithm gives a solution that is on expectation at least 0.878 times the optimal solution in (3.11).

$$\begin{aligned} \underset{\mathbf{x}}{\text{maximize}} \quad & \frac{1}{2} \sum_{i<j} A_{i,j}(1 - \mathbf{v}_i \mathbf{v}_j) \\ \text{subject to} \quad & \mathbf{v}_i^\top \mathbf{v}_i = 1 \text{ for } i = 1, \ldots, n. \end{aligned} \tag{3.12}$$

Problem (3.12) is a relaxed version of the problem (3.11), since the vector $\mathbf{v}_i$ can be regarded as x_i extended into higher dimension. To relate the problem to the semidefinite programming, Problem (3.11) can be reformulated according to Lemma 3.4.

Lemma 3.4. *Let $\mathbf{L}$ be the unnormalized Laplacian matrix of G, with $\mathbf{L} = \mathbf{D} - \mathbf{A}$. Problem (3.11) has an equivalent form as problem (3.13).*

$$\begin{aligned}
\underset{\mathbf{X}}{\text{maximize}} \quad & \frac{1}{4}\langle \mathbf{L}, \mathbf{X} \rangle \\
\text{subject to} \quad & diag(\mathbf{X}) = \mathbf{e}, \\
& rank(\mathbf{X}) = 1, \\
& \mathbf{X} \succeq 0.
\end{aligned} \tag{3.13}$$

Proof. We prove by showing the equivalences between the conditions and the objective function. $\mathbf{X} \succeq 0$ and $rank(\mathbf{X}) = 1$ if and only if there exists a vector $\mathbf{x}$ such that $\mathbf{X} = \mathbf{x}\mathbf{x}^\top$; besides, for the decomposed $\mathbf{x}$, $diag(\mathbf{X}) = \mathbf{e}$ if and only if $\mathbf{x}_i^2 = 1, i = 1, 2, \ldots, n$. These show the equivalence between conditions. Since $\frac{1}{4}\langle \mathbf{L}, \mathbf{X} \rangle = \frac{1}{4}\mathbf{x}^\top L\mathbf{x} = \frac{1}{4}\sum_i \sum_j A_{i,j} - \frac{1}{4}\mathbf{x}^\top A_{i,j}\mathbf{x} = \frac{1}{2}\sum_{i<j} A_{i,j}(1 - x_i x_j)$, the objective function is the same as well. □

According to Corollary 3.4.1, we show that the relaxed Max Cut problem is actually semidefinite relaxation.

Corollary 3.4.1. *Problem (3.12) has an equivalent form as Problem (3.14).*

$$\begin{aligned}
\underset{\mathbf{X}}{\text{maximize}} \quad & \frac{1}{4}\langle \mathbf{L}, \mathbf{X} \rangle \\
\text{subject to} \quad & diag(\mathbf{X}) = \mathbf{e}, \\
& \mathbf{X} \succeq 0.
\end{aligned} \tag{3.14}$$

Proof. $\mathbf{X} \succeq 0$ if and only if there exists a matrix $\mathbf{V}$ of n columns such that $\mathbf{V}^\top\mathbf{V} = \mathbf{X}$. Let the ith column vector of $\mathbf{V}$ be $\mathbf{v}_i$; $diag(\mathbf{X}) = \mathbf{e}$ if and only if $\mathbf{v}_i^\top \mathbf{v}_i = 1$ for $i = 1, 2, \ldots, n$. Thus the condition part is equivalent. On the other hand,

$$\begin{aligned}
\frac{1}{4}\langle \mathbf{L}, \mathbf{X} \rangle &= \frac{1}{4}\sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}\mathbf{v}_i^\top \mathbf{v}_i - \frac{1}{4}\sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}\mathbf{v}_i^\top \mathbf{v}_j \\
&= \frac{1}{4}\sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}(\mathbf{v}_i^\top \mathbf{v}_i - \mathbf{v}_i^\top \mathbf{v}_j) \\
&= \frac{1}{2}\sum_{i<j} A_{ij}(1 - \mathbf{v}_i^\top \mathbf{v}_j).
\end{aligned}$$

□

Thus the objective function is equivalent as well.

3.4.2 Quadratic Knapsack Problem

Employing Semidefinite Programming to solve the Max Cut Problem has inspired many works. The one most closely connected to the thesis is using the semidefinite programming to solve the Quadratic Knapsack Problem (QKP). Let $\mathbf{x}$ be an $n \times 1$

vector variable, $\mathbf{P}$ be any $n \times n$ real matrix, $\mathbf{b}$ be an $n \times 1$ vector of non-negative entries, QKP has the form in (QKP). The problem is different with the Max Cut problem in three aspects. First, unlike the graph Laplacian matrix, $\mathbf{P}$ is not necessarily positive semidefinite; Second, $\mathbf{x}$ is bounded by a linear constraint; Third, the feasible area of variable $\mathbf{x}$ is different.

$$\begin{aligned} \underset{\mathbf{x}}{\text{maximize}} \quad & \mathbf{x}^\top \mathbf{P} \mathbf{x} \\ \text{subject to} \quad & \mathbf{b}^\top \mathbf{x} \leq k, \\ & \mathbf{x} \in \{0,1\}^n. \end{aligned} \tag{QKP}$$

The Quadratic Knapsack Problem (QKP) is a generalized version of the Knapsack Problem (KP), it can be reduced to KP when $\mathbf{P}$ is a diagonal matrix with non-negative entries. Intuitively, the QKP should be harder than KP. To put it clearly, we introduce a decision setting of both problems, denoted as KP-decision and QKP-decision. Take the problem QKP-decision as an example. Given any feasible solution QKP, the QKP-decision calculates whether there exists a better feasible solution; the optimal solution can be found by repeating the QKP-decision and binary search in $\mathrm{O}(\log(\sum_{i,j} |P_{i,j}|))$ times. Kellerer *et al.* [20] gives an introduction to the time complexity of a family of knapsack problems. It concludes that both the KP-decision and the QKP-decision belong to the NP-Complete problem class. The KP-decision as a weakly NP-Hard problem, is solvable in pseudopolynomial time. Nevertheless, the QKP-decision belongs to the strongly NP-Hard problems, which can be regarded as a reduction from the the Clique Problem, as one of the Karp's 21 NP-complete problems [37].

Remark. *The Clique Problem can be formulated as a quadratic knapsack problem.*

Proof. The Clique Problem can be stated as Kellerer *et al.* [20]: Given a positive integer k, and an undirected graph $G = (V, E)$, check whether G contains a complete subgraph on k nodes. We can use the notation in Equation (QKP) here, let $P_{i,j} = P_{j,i} = 1$ if $(i,j) \in E$, otherwise, assign the value of the entries by 0. $\mathbf{b}$ be a vector of all ones. Then G contains the complete subgraph if and only if the optimal value of the QKP is $k(k-1)$. □

The following theorem [38] proves that when $\mathbf{P}$ contains both positive and negative entries, the QKP are hard even in approximation sense.

Theorem 3.5 (Rader *et al.* [38], Pisinger [39])**.** *QKP with positive and negative coefficients does not have any polynomial time approximation algorithm with fixed approximation ratio unless $P = NP$.*

One way to solve the hard QKP is to give it reasonable relaxations. The survey [39] provides us with an overview of several techniques to derive a upper bound. For instance, the Linearization [40, 41, 42] which transforms the QKP into a KP problem ; the Lagrangian relaxation [43]; the Lagrangian decomposition [44, 45] which transforms the QKP into two or more independent subproblems, and the semidefinite relaxation [21].

We employ the semidefinite relaxation method [21]. While four semidefinite relaxation methods are discussed, we specifically describe the one with the best trade-off between time and quality. For the convenience of further analysis, we first reformulate problem (QKP) according to Lemma 3.6.

Lemma 3.6. *Problem (3.15) is an equivalent form of Problem (QKP).*

$$\begin{aligned} \underset{\mathbf{x}}{\textit{maximize}} \quad & \langle \mathbf{P}, \mathbf{X} \rangle \\ \textit{subject to} \quad & \langle \mathbf{b}\mathbf{b}^\top, \mathbf{X} \rangle \le k^2, \\ & \textit{rank}(\mathbf{X}) = 1, \\ & \mathbf{X} \in \{0,1\}^{n \times n}. \end{aligned} \tag{3.15}$$

Proof. The condition $\textit{rank}(\mathbf{X}) = 1$ and $\mathbf{X} \in \{0,1\}^{n\times n}$ are equivelent with the condition that there exists an $n \times 1$ 0-1 vector $\mathbf{x}$ such that $\mathbf{x}\mathbf{x}^\top = \mathbf{X}$. Based on the decomposed $\mathbf{X}$, $\langle \mathbf{b}\mathbf{b}^\top, \mathbf{X}\rangle \le k^2$ is equivalent with $(\mathbf{b}^\top\mathbf{x})^2 \le k^2$, and $\langle \mathbf{P}, \mathbf{X}\rangle$ is equivalent with $\mathbf{x}^\top\mathbf{P}\mathbf{x}$. Since the elements of $\mathbf{b}$ and $\mathbf{x}$ are non-negative, $(\mathbf{b}^\top\mathbf{x})^2 \le k^2$ is equivalent with $\mathbf{b}^\top\mathbf{x} \le k$. Putting together, we show that both the conditions and the objective function are equivalent between problem (QKP) and the problem (3.15). $\square$

However, it is hard to tackle the binary constraint. Following what has been done in the Max Cut problem, we relax the binary constraint into $0 \le X_{i,j} \le 1$ for $i = 1, \ldots, n$, $j = 1, \ldots, n$. In Lemma 3.7 we relax the rank constraint and the binary constraint.

Lemma 3.7. *For any $n \times n$ matrix $\mathbf{X}$ whose rank equals to one, if all of its entries are bounded by $0 \le X_{i,j} \le 1$, then:*

$$\mathbf{X} \succeq \textit{diag}(\mathbf{X})\textit{diag}(\mathbf{X})^\top.$$

Proof. Let $\mathbf{X} = \mathbf{v}\mathbf{v}^\top$, $\textit{diag}(\mathbf{X})_i = \mathbf{v}_i^2$, $\mathbf{v}_i^2 \le \mathbf{v}_i$ since $0 \le \mathbf{v}_i \le 1$, let $\mathbf{a}$ be any vector, $\mathbf{a}^\top(\mathbf{X} - \textit{diag}(\mathbf{X})\textit{diag}(\mathbf{X})^\top)\mathbf{a} = (\sum_{i=1}^{n} \mathbf{a}_i\mathbf{v}_i)^2 - (\sum_{i=1}^{n} \mathbf{a}_i\mathbf{v}_i^2)^2 \ge 0$. $\square$

Lemma 3.8. *Problem (QKP) can be relaxed into problem (QKP-SDP).*

$$\begin{aligned} \underset{\mathbf{x}}{\textit{maximize}} \quad & \langle \mathbf{P}, \mathbf{X} \rangle \\ \textit{subject to} \quad & \langle \mathbf{b}\mathbf{b}^\top, \mathbf{X} \rangle \le k^2, \\ & \begin{bmatrix} \mathbf{X} & \textit{diag}(\mathbf{X}) \\ \textit{diag}(\mathbf{X})^\top & 1 \end{bmatrix} \succeq 0. \end{aligned} \tag{QKP-SDP}$$

Proof. According to the Lemma 3.7, we learn that the rank constraint and the binary constraint together can be relaxed into $\mathbf{X} - \textit{diag}(\mathbf{X})\textit{diag}(\mathbf{X})^\top \succeq 0$, the left hand side of which is the Schur complement of 1 of the large composed matrix $\begin{bmatrix} \mathbf{X} & \textit{diag}(\mathbf{X}) \\ \textit{diag}(\mathbf{X})^\top & 1 \end{bmatrix}$. The expression $\mathbf{X} - \textit{diag}(\mathbf{X})\textit{diag}(\mathbf{X})^\top$ is positive semidefinite if and only if the large matrix is positive semidefinite. Thus we can relax the problem (QKP) into the semidefinite programming problem (QKP-SDP). $\square$

3.4.3 Upper Bound

Consider a special situation that $\mathbf{b} = \mathbf{e}$, we can derive an upper bound of the (QKP-SDP) as follows. With the coefficients $\lambda \geq 0$, and $\gamma \geq 0$, the Lagrangian function of (QKP-SDP) can be written as:

$$\begin{aligned}\mathcal{L}(\mathbf{X}, \lambda, \gamma) &= \langle \mathbf{P}, \mathbf{X} \rangle + \lambda(k^2 - \langle \mathbf{e}\mathbf{e}^\top, \mathbf{X} \rangle) + \gamma \lambda_{min} \langle \mathbf{X} - diag(\mathbf{X}) diag(\mathbf{X})^\top \rangle \\ &= \lambda k^2 + \langle \mathbf{P} - \lambda \mathbf{e}\mathbf{e}^\top, \mathbf{X} \rangle + \min_{\mathbf{Y} \succeq 0, trace(\mathbf{Y}) = \gamma} (\mathbf{Y}, \mathbf{X} - diag(\mathbf{X}) diag(\mathbf{X})^\top) \\ &= \lambda k^2 + \min_{\mathbf{Y} \succeq 0, trace(\mathbf{Y}) = \gamma} \langle \mathbf{P} - \lambda \mathbf{e}\mathbf{e}^\top + \mathbf{Y}, \mathbf{X} \rangle + \langle \mathbf{Y}, -diag(\mathbf{X}) diag(\mathbf{X})^\top \rangle.\end{aligned} \tag{Lagrangian}$$

The corresponding dual function is:

$$\begin{aligned} g(\lambda, \gamma) &= \max_{\mathbf{X}} \mathcal{L}(\mathbf{X}, \lambda, \gamma) \\ &= \lambda k^2 + \max_{\mathbf{X}} \min_{\mathbf{Y} \succeq 0, trace(\mathbf{Y}) = \gamma} \langle \mathbf{P} - \lambda \mathbf{e}\mathbf{e}^\top + \mathbf{Y}, \mathbf{X} \rangle + \langle \mathbf{Y}, -diag(\mathbf{X}) diag(\mathbf{X})^\top \rangle \\ &\leq \lambda k^2 + \min_{\mathbf{Y} \succeq 0, trace(\mathbf{Y}) = \gamma} \max_{\mathbf{X}} \langle \mathbf{P} - \lambda \mathbf{e}\mathbf{e}^\top + \mathbf{Y}, \mathbf{X} \rangle \\ &= \lambda k^2 + \min_{\mathbf{Y} \succeq 0, trace(\mathbf{Y}) = \gamma} \begin{cases} 0 & \text{if} \quad \mathbf{P} - \lambda \mathbf{e}\mathbf{e}^\top + \mathbf{Y} = 0 \\ +\infty & \text{if} \quad \text{otherwise} \end{cases} \\ &= \begin{cases} \lambda k^2 & \text{if} \quad -\lambda \mathbf{e}\mathbf{e}^\top + \mathbf{P} \preceq 0 \\ +\infty & \text{if} \quad \text{otherwise.} \end{cases} \end{aligned} \tag{Dual Function}$$

The dual problem is:

$$\begin{aligned} \underset{\lambda}{\text{minimize}} \quad & \lambda k^2 \\ \text{subject to} \quad & \lambda \mathbf{e}\mathbf{e}^\top \succeq \mathbf{P}, \end{aligned} \tag{Dual SDP}$$

where $\lambda \mathbf{e}\mathbf{e}^\top - \mathbf{P} \succeq 0$ means that $\min_{\mathbf{v}, \mathbf{v}^\top \mathbf{v} = 1} \mathbf{v}^\top \lambda \mathbf{e}\mathbf{e}^\top \mathbf{v} - \mathbf{v}^\top \mathbf{P} \mathbf{v} \geq 0$, $\lambda \geq \min_{\mathbf{v}} \frac{\mathbf{v}^\top \mathbf{P} \mathbf{v}}{(\sum_{i=1}^n \mathbf{v}_i)^2}$. As a result, we could find an upper bound as $\min_{\mathbf{v}} \frac{\mathbf{v}^\top \mathbf{P} \mathbf{v}}{(\sum_{i=1}^n \mathbf{v}_i)^2} k^2$.

Besides, $\lambda_{\max}(\mathbf{P}) = \max_{\mathbf{v}} \frac{\mathbf{v}^\top \mathbf{P} \mathbf{v}}{\mathbf{v}^\top \mathbf{v}} \geq \frac{\mathbf{x}^\top \mathbf{P} \mathbf{x}}{\mathbf{x}^\top \mathbf{x}} = \frac{\mathbf{x}^\top \mathbf{P} \mathbf{x}}{k}$, thus $\mathbf{x}^\top \mathbf{P} \mathbf{x} \leq k \lambda_{\max}(\mathbf{P})$. $k \lambda_{\max} \mathbf{P}$ could be another upper bound.

3.4.4 Rounding Techniques

(QKP-SDP) gave us an upper bound of the Quadratic Knapsack Problem (QKP), however, there is no clue how to extract the vector $\mathbf{x}^*$ out the optimal matrix $\mathbf{X}^*$. Ideally, if $rank(\mathbf{X}^*) = 1$, then we could decompose $\mathbf{X}^*$ into $\mathbf{x}^*\mathbf{x}^{*\top}$. In applications, the rank of $\mathbf{X}^*$ is larger than one. Unlike the Max Cut problem, in which the random projection could be applied based on the fact that $diag(\mathbf{X}) = \mathbf{e}$. Different than the random projection scheme, the Gaussian Randomization Procedure [22] can be adapted to solve our problem in Algorithm 1. t is a parameter regarding the number of iterations, usually we can set it as 1000.

Algorithm 1 Gaussian Randomization Procedure

```
input  : X*, n, k, t
output : x*
Form covariance matrix Σ ← X* − diag(X*) diag(X*)^⊤
for i ← 1, ..., t do
    sample x_0 ~ N(diag(X*), Σ)  do
        x' ← randomized_rounding(x_0)
    while b^⊤ x' > k;
    if f < x'P x' then
        x ← x' and f ← x'P x'
return x
```

The randomized_rounding is a subrountine with:

$$\mathbb{P}(\text{randomized\_rounding}(\mathbf{x})[i] = 1) = \max\{\min\{1, x_i\}, 0\}.$$

4 Bounded-box Diversity Maximization

As we have discussed in the background Section 2.1, Antonis *et al.* [18] assume that the users' *exposure index* is binary. Users may either always be exposed to news about the Clinton Foundation receiving donations from Middle-Eastern countries or Donald Trump's involvement with Russian Election interference, but not both. Although the recommendation system reinforces user confirmation bias, it won't be as extreme as totally censoring the information. More realistically, the overall exposure should be a bounded continuous value. In this chapter, we change the formulation and solve a continuous diversity maximization problem.

4.1 Notations

Let $G = (V, E, w)$ be a simple weighted graph that represents the network. Respectively, denoting by V the set of users, by E the set of connections between the users, and by function $w(i, j)$ the connection strength of node i and j. Let $\mathbf{A}$ be the adjacency matrix of G, let $\mathbf{D}$ be a diagonal matrix, with $D_{i,i} = \sum_j A_{i,j}$. Let $\mathbf{L}$ be the unnormalized Laplacian matrix of G. $\mathbf{L} = \mathbf{D} - \mathbf{A}$.

Let $\mathbf{X}$ and $\mathbf{Y}$ be matrices of size $n \times n$. Let $\mathbf{x}$ be a vector of size n. Denote by $\langle \mathbf{X}, \mathbf{Y} \rangle$ the Frobenius inner product of two matrix $\mathbf{X}$ and $\mathbf{Y}$, $\langle \mathbf{X}, \mathbf{Y} \rangle = \sum_{i=1}^{n} \sum_{j=1}^{n} X_{ij} Y_{ij}$. Denote by $diag(\mathbf{X})$ the diagonal vector of $\mathbf{X}$, $diag(\mathbf{X}) = [X_{11}, X_{22}, \dots, X_{nn}]^\top$. Denote by $Diag(\mathbf{X})$ a diagonal matrix whose diagonal equals to $diag(\mathbf{X})$, denote by $Diag(\mathbf{x})$ a diagonal matrix whose diagonal equals to $\mathbf{x}$. Denote by $rank(\mathbf{X})$ the rank of $\mathbf{X}$. Let $\mathbf{x}$ be a vector, let $card(\mathbf{x})$ be the number of non-zero elements in $\mathbf{x}$. Let $\mathbf{e}$ be a vector of all ones.

4.2 Problem Formulation

Recall that the *exposure index* measures overall opinion that a user is exposed to in the social network. Instead of a binary form (either -1 or 1), we let the *exposure index* be a continuous value bounded in $[-1, 1]$. We keep the definition *diversity maximization* as in Definition 2.1, which measures the overall connections between the opposite opinion exposures in the social network, denoted by $\eta(G, \mathbf{s}) = \sum_{(i,j)\in E} W(i, j)(s_i - s_j)^2$. Our task is to maximize the *diversity index* in a given graph by changing a given number of *exposure indexes*. Consistently with this definition, the initial state as well as the final state of the *exposure indexes* are values between -1 and 1. We formulate the continuous diversity maximization problem in Problem 4.1.

Problem 4.1 (Diversity Maximization)**.** *Given a graph $G = (V, E, w)$ and an exposure index vector $\mathbf{s}$ with $s_i \in [-1, 1], i = 1, \ldots, n$; change at most k elements in $\mathbf{s}$ to get a new exposure index $\mathbf{y}$ with $y_i \in [-1, 1], i = 1, \ldots, n$, such that the new diversity index $\eta(G, \mathbf{y})$ is maximized.*

The Problem 4.1 has a direct mathematical programming form as Problem (4.1).

$$\begin{aligned} \underset{\mathbf{y}}{\text{maximize}} \quad & \eta(G, \mathbf{y}) \\ \text{subject to} \quad & ||\mathbf{y}||_\infty \leq 1, \\ & card(\mathbf{y} - \mathbf{s}) \leq k. \end{aligned} \tag{4.1}$$

We noticed that $\eta(G, \mathbf{y}) = \sum_{(i,j)\in E} A_{ij}(y_i - y_j)^2 = \mathbf{y}^\top \mathbf{D}\mathbf{y} - \mathbf{y}^\top \mathbf{A}\mathbf{y} = \mathbf{y}^\top \mathbf{L}\mathbf{y}$, where the $\mathbf{L}$ is the Laplacian matrix of G. The more important attribute of $\mathbf{L}$ is that it is a positive semidefinite matrix. Without the cardinality constraint, Problem (4.1) is a convex maximization problem, as we discussed in the Chpater 2, the optimal solution is reached when $\mathbf{y}$ is assigned by the extreme values, with the box bounding constraint $\mathbf{y} \in \{-1, 1\}^n$. In Lemma 4.2 we show that with the cardinality constraint, $y_i \in \{-1, 1, s_i\}, i = 1, 2, \ldots, n$.

Lemma 4.2. *Let $\mathbf{y}$ be the optimal solution of Problem (4.1), $y_i \in \{-1, 1, s_i\}, i = 1, 2, \ldots, n$.*

Proof. We write $\mathbf{y}^\top$ as $[\mathbf{y}_1^\top \; y \; \mathbf{y}_2^\top]$, where y is any element in $\mathbf{y}$. Let ℓ be the coefficient of y^2 in the polynomial $\mathbf{y}^\top \mathbf{L}\, \mathbf{y}$. Then ℓ is one of the elements at the diagonal of the Laplacian $\mathbf{L}$, and $\mathbf{y}^\top \mathbf{L}\mathbf{y}$ can be written as $C_1 + C_2 y + \ell y^2$. Here C_1 and C_2 are independent of y. Since ℓ is a diagonal element of $\mathbf{L}$, it is non-negative. It follows that $\arg \max_{y\in[-1,1]} \mathbf{y}^\top \mathbf{L}\, \mathbf{y}$ is either 1 or -1. □

According to Lemma 4.2, Problem (4.1) can be written equivalently as Problem (4.2).

$$\begin{aligned} \underset{\mathbf{y}}{\text{maximize}} \quad & \mathbf{y}^\top L\mathbf{y} \\ \text{subject to} \quad & y_i \in \{-1, 1, s_i\}, i = 1, 2, \ldots, n, \\ & card(\mathbf{x}) \leq k, \\ & \mathbf{x} = \mathbf{y} - \mathbf{s}. \end{aligned} \tag{4.2}$$

The hardness of Problem (4.2) is in three aspects. First, the objective function is a convex maximization function, ; second, each element of the variable $\mathbf{y}$ has three choices; third, a cardinality constraint is put on an affine function of the variable. The form of the Problem (4.2) is similar with the Quadratic Knapsack Problem (QKP) we define in Section 3.4.2 from the persepetive of the discrete varaibles, the cardinality constraint and the form of objective function. To formulate Problem (4.2) into a Quadratic Knapsack Problem, we first introduces two auxiliary variables.

Lemma 4.3. *Let* $\mathbf{s_a} = \mathbf{e} - \mathbf{s}$, $\mathbf{s_b} = -\mathbf{e} - \mathbf{s}$. *Problem* (4.3) *is equivalent with Problem* (4.2).

$$\begin{aligned}
\underset{\mathbf{a},\mathbf{b}}{\textit{maximize}} \quad & (\mathbf{s} + \mathit{Diag}(\mathbf{s_a})\mathbf{a} + \mathit{Diag}(\mathbf{s_b})\mathbf{b})^\top L(\mathbf{s} + \mathit{Diag}(\mathbf{s_a})\mathbf{a} + \mathit{Diag}(\mathbf{s_b})\mathbf{b}) \\
\textit{subject to} \quad & \mathbf{a} \in \{0,1\}^n, \\
& \mathbf{b} \in \{0,1\}^n, \\
& a_i + b_i \neq 2, i = 1, 2, \dots, n, \\
& \mathbf{e}^\top(\mathbf{a} + \mathbf{b}) \leq k.
\end{aligned} \tag{4.3}$$

Proof. First, according to the conditions $\mathbf{a} \in \{0,1\}^n$, $\mathbf{b} \in \{0,1\}^n$ and $a_i + b_i \neq 2$, (a_i, b_i) can be $(0,1)$, $(1,0)$ or $(1,1)$. Thus,

$$(\mathbf{s} + \mathit{Diag}(\mathbf{s}_a)\mathbf{a} + \mathit{Diag}(\mathbf{s}_b)\mathbf{b})_i = \begin{cases} s_i & \text{if} \quad a_i = 0, b_i = 0, \\ 1 & \text{if} \quad a_i = 1, b_i = 0, \\ -1 & \text{if} \quad a_i = 0, b_i = 1. \end{cases}$$

Second, the conditions that $a_i + b_i \neq 2$ and $\mathbf{e}^\top(\mathbf{a} + \mathbf{b}) \leq k$ imply that the number of 1s in the vector $\mathbf{a}+\mathbf{b}$ is less than k, thus the number of 1s and -1s in $(\mathbf{s}+\mathit{Diag}(\mathbf{s}_a)\mathbf{a}+\mathit{Diag}(\mathbf{s}_b)\mathbf{b})$ is less than k.

Third, we can use a vector $\mathbf{y}$ to replace the $(\mathbf{s} + \mathit{Diag}(\mathbf{s}_a)\mathbf{a} + \mathit{Diag}(\mathbf{s}_b)\mathbf{b})$, and condition it with $\mathbf{y} \in \{-1, 1, s_i\}^n$ and $\mathit{card}(\mathbf{y} - \mathbf{s}) \leq k$. The conditions and the objective function are identical with the Problem (4.2). $\square$

We first observe that the inequality constraint can be dropped without changing the optimal solution of the Problem (4.3).

Corollary 4.3.1. *Dropping the inequality constraint* $(a_i+b_i \neq 2,$ *for all* $i = 1, \dots, n)$ *does not influence the optimal solution of Problem* (4.3).

Proof. Consider a solution with $a_i + b_i = 2$ for some $i = 1, \dots, n$. Since a_i and b_i are binary, we have $a_i = b_i = 1$. Since $\mathbf{x} = \mathit{Diag}(\mathbf{s_a})\,\mathbf{a} + \mathit{Diag}(\mathbf{s_b})\,\mathbf{b}$, we have $x_i = s_{ai} + s_{bi} = -2s_i$, which means that s_i is changed to $-s_i$. Furthermore, such a solution contributes exactly 2 to the cardinality of $\mathbf{a} + \mathbf{b}$. However, as proved in Lemma 4.2, if s_i is changed in the optimal condition, then it should be changed to either 1 or -1. We conclude that $a_i + b_i \neq 2$ will not be violated in the optimal solution, and thus, it can be dropped. $\square$

Let $\mathbf{v}$ be a binary vector of dimension $2n$ that concatenates vectors $\mathbf{a}$ and $\mathbf{b}$, i.e., $\mathbf{v}^\top = [\mathbf{a}^\top | \mathbf{b}^\top]$. Let $\mathbf{q}^\top = [\mathbf{s}^\top \mathbf{L}\mathit{Diag}(\mathbf{s_a}) | \mathbf{s}^\top \mathbf{L}\mathit{Diag}(\mathbf{s_b})]$,

$$\mathbf{P} = \begin{pmatrix} \mathit{Diag}(\mathbf{s_a})\mathbf{L}\mathit{Diag}(\mathbf{s_a}) & \mathit{Diag}(\mathbf{s_a})\mathbf{L}\mathit{Diag}(\mathbf{s_b}) \\ \mathit{Diag}(\mathbf{s_b})\mathbf{L}\mathit{Diag}(\mathbf{s_a}) & \mathit{Diag}(\mathbf{s_b})\mathbf{L}\mathit{Diag}(\mathbf{s_b}) \end{pmatrix}.$$

Lemma 4.4. *Problem (4.4) is equivalent with Problem (4.2).*

$$\begin{aligned} \underset{\mathbf{v}}{\textit{maximize}} \quad & \mathbf{v}^\top \mathbf{P}\mathbf{v} + 2\mathbf{q}^\top \mathbf{v} + c_1 \\ \textit{subject to} \quad & \mathbf{v} \in \{0,1\}^{2n}, \\ & \mathbf{e}^\top \mathbf{v} \leq k. \end{aligned} \tag{4.4}$$

Proof. According to Lemma 4.3, we only need to show that Problem (4.4) is equivalent with Problem (4.3). The condition part is equivalent, according to the construction of $\mathbf{v}$ and the Corollary 4.3.1.

Denote $\mathbf{s}^\top \mathbf{L} \mathit{Diag}(\mathbf{s}_a)$ by $\mathbf{c}_a^\top$, denote $\mathbf{s}^\top \mathbf{L} \mathit{Diag}(\mathbf{s}_b)$ by $\mathbf{c}_b^\top$, denote $\mathit{Diag}(\mathbf{s}_a)\mathbf{L}\mathit{Diag}(\mathbf{s}_b)$ by $\mathbf{C}_{ab}$, denote $\mathit{Diag}(\mathbf{s}_a)\mathbf{L}\mathit{Diag}(\mathbf{s}_a)$ by $\mathbf{C}_{aa}$, denote $\mathit{Diag}(\mathbf{s}_b)\mathbf{L}\mathit{Diag}(\mathbf{s}_b)$ by $\mathbf{C}_{bb}$, denote $\mathbf{s}^\top \mathbf{L}\mathbf{s}$ by c_1. The objective function of the Problem (4.3) can be written as:

$$c_1 + 2\mathbf{c}_a^\top \mathbf{a} + 2\mathbf{c}_b^\top \mathbf{b} + 2\mathbf{a}^\top \mathbf{C}_{ab}\mathbf{b} + \mathbf{a}^\top \mathbf{C}_{aa}\mathbf{a} + \mathbf{b}^\top \mathbf{C}_{bb}\mathbf{b}. \tag{4.5}$$

Let $\mathbf{v}$ be a $2n \times 1$ vector, $\mathbf{v}^\top = [\mathbf{a}^\top | \mathbf{b}^\top]$, let $\mathbf{P} = \begin{pmatrix} \mathbf{C}_{aa} & \mathbf{C}_{ab} \\ \mathbf{C}_{ab}^\top & \mathbf{C}_{bb} \end{pmatrix}$, let $\mathbf{q}^\top = [\mathbf{c}_a^\top | \mathbf{c}_b^\top]$. Then Equation (4.5) is equivalent with Equation (4.6).

$$\mathbf{v}^\top \mathbf{P}\mathbf{v} + 2\mathbf{q}^\top \mathbf{v} + c_1. \tag{4.6}$$

□

Problem 4.5 is a Quadratic Knapsack Problem. The quadratic Knapsack problem belongs to the strongly NP-Hard problems [20], and can not find a polynomial time approximation algorithm with fixed approximation ratio [39] unless P = NP. In the following part of this Chapter we discuss some of the heuristics that can be employed to solve this problem in practice.

4.3 Semidefinite Relaxation

Introduce a $2n \times 2n$ matrix $\mathbf{V}$. Let $\tilde{\mathbf{P}} = \begin{pmatrix} \mathbf{P} & \mathbf{q} \\ \mathbf{q}^\top & c_1 \end{pmatrix}$. According to Lemma 4.5, we can first formulate Problem (4.5) similarly with the Problem (3.15), and then apply the Semidefinite Relaxation as Problem (QKP-SDP).

Lemma 4.5. *Problem (4.7) is an equivalent form of Problem (4.4).*

$$\begin{aligned} \underset{\tilde{\mathbf{V}}}{\textit{maximize}} \quad & \langle \tilde{\mathbf{P}}, \tilde{\mathbf{V}} \rangle \\ \textit{subject to} \quad & \tilde{\mathbf{V}} = \begin{pmatrix} \mathbf{V} & \mathbf{v} \\ \mathbf{v}^\top & 1 \end{pmatrix}, \\ & \langle \mathbf{e}\mathbf{e}^\top, \mathbf{V} \rangle \leq k^2, \\ & \mathit{rank}(\mathbf{V}) = 1, \\ & \tilde{\mathbf{V}} \in \{0,1\}^{(2n+1)\times(2n+1)}. \end{aligned} \tag{4.7}$$

Proof. The proof is the same as Lemma 3.6, except that there is a constant and a affine function in the objective function. As a result, we introduced the block matrix $\tilde{\mathbf{P}}$ and $\tilde{\mathbf{V}}$, and the objective function can be written equivalently as the Frobenius inner product of $\tilde{\mathbf{P}}$ and $\tilde{\mathbf{V}}$, namely $\langle \tilde{\mathbf{P}}, \tilde{\mathbf{V}} \rangle$. □

There is a implicit information about the $\mathbf{V}$ that it can be represented as $\mathbf{V} = \mathbf{v}\mathbf{v}^\top$, since $\mathbf{v}$ is a 0-1 vector, it can be represented as $\mathbf{V} = diag(\mathbf{V})diag(\mathbf{V})^\top$. As is discussed in section 3.4.2, the condition $\mathbf{V} = diag(\mathbf{V})diag(\mathbf{V})^\top$ and $\mathbf{V} \in \{0,1\}^{2n\times 2n}$ together can be relaxed into $\mathbf{V} \succcurlyeq diag(\mathbf{V})diag(\mathbf{V})^\top$.

Lemma 4.6. *Problem (4.8) is a relaxed form of Problem (4.4).*

$$\begin{aligned} \underset{\tilde{\mathbf{V}}}{\textit{maximize}} \quad & \langle \tilde{\mathbf{P}}, \tilde{\mathbf{V}} \rangle \\ \textit{subject to} \quad & \tilde{\mathbf{V}} = \begin{pmatrix} \mathbf{V} & \mathbf{v} \\ \mathbf{v}^\top & 1 \end{pmatrix}, \\ & \tilde{\mathbf{V}} \succcurlyeq 0, \\ & diag(\mathbf{V}) = \mathbf{v}, \\ & \langle \mathbf{e}\mathbf{e}^\top, \mathbf{V} \rangle \leq k^2. \end{aligned} \tag{4.8}$$

Notice that instead of modeling $\mathbf{e}^\top\mathbf{v} \leq k$ with $\langle \mathbf{e}\mathbf{e}^\top, \mathbf{V} \rangle \leq k^2$, we replace it with a tighter condition $\langle \mathbf{e}\mathbf{e}^\top - \mathbf{I}, \mathbf{V} \rangle \leq k^2 - k$ according to Corollary 4.6.1.

Corollary 4.6.1 (Helmberg *et al.* [21]). *Condition $\langle \mathbf{e}\mathbf{e}^\top - \mathbf{I}, \mathbf{V} \rangle \leq k^2 - k$ is tighter than $\langle \mathbf{e}\mathbf{e}^\top, \mathbf{V} \rangle \leq k^2$ in the Problem (4.8).*

Proof. We have $\mathbf{e}^\top\mathbf{v} \leq k$ and $\langle \mathbf{e}\mathbf{e}^\top\mathbf{V} \rangle - \mathbf{e}^\top\mathbf{v} \leq k^2 - k$, implying $\langle \mathbf{e}\mathbf{e}^\top, \mathbf{V} \rangle \leq k^2$; thus $\langle \mathbf{e}\mathbf{e}^\top\mathbf{V} \rangle - \mathbf{e}^\top\mathbf{v} \leq k^2 - k$ is a tighter condition than $\langle \mathbf{e}\mathbf{e}^\top, \mathbf{V} \rangle \leq k^2$. On the other hand, in optimal cases of the original problem, $\mathbf{v}^\top\mathbf{e} \leq k$, $\mathbf{v}^\top\mathbf{e} - 1 \leq k - 1$ and $\mathbf{V} = \mathbf{v}\mathbf{v}^\top$ imply $\langle \mathbf{e}\mathbf{e}^\top, \mathbf{V} \rangle - \mathbf{e}^\top\mathbf{v} \leq k^2 - k$, thus the latter condition includes the optimal solution. Since $diag(\mathbf{V}) = \mathbf{v}$, $\langle \mathbf{e}\mathbf{e}^\top, \mathbf{V} \rangle - \mathbf{e}^\top\mathbf{v}$ could be written as $\langle \mathbf{e}\mathbf{e}^\top - \mathbf{I}, \mathbf{V} \rangle$. □

4.3.1 Rounding Techniques

Assume that we have solved Problem (4.8) and got the optimal solution $\tilde{\mathbf{V}}^*$ of the Problem (4.8). Let $\tilde{\mathbf{V}}^* = \begin{pmatrix} \mathbf{V}^* & \mathbf{v}^* \\ \mathbf{v}^{*\top} & 1 \end{pmatrix}$. Ideally, $\mathbf{v}^*\mathbf{v}^{*\top} = \mathbf{V}^*$ and $\mathbf{v}^*$is the optimal solution of the Problem (4.4). While by solving the Problem (4.8), $\mathbf{v}^*$ is continuous and $\mathbf{V}^* \neq \mathbf{v}^*\mathbf{v}^{\top *}$.

We can either recover $\mathbf{v}$ of Problem (4.4) directly from $\mathbf{v}^*$, or from $\mathbf{v}^*$ and $\mathbf{V}^*$ together. For the first method, in order to get the discrete version of the $\mathbf{v}^*$, we can apply a heuristic method to assume that $v_i = 1$ with probability v_i^* and with probability $1 - v_i^*$, $v_i = 0$. This method does not work well in experiments.

The second rounding technique we use is from Luo *et al.* [22], which provides us with a randomized rounding technique: Gaussian Randomization Procedure to solve the problem. Based on this relaxed equations, the $\mathbf{V}^*$ we obtain is not a rank one matrix, as a result, we can not extract the feasible $\mathbf{v}^*$ for Problem (4.4) directly by assuming that $\mathbf{V}^* = \mathbf{v}^*\mathbf{v}^{*\top}$. But we can randomly draw a vector $\mathbf{a}$ in a Multivariate Gaussian Distribution, with $\mathbf{v}^*$ as the mean and $\mathbf{V}^* - \mathbf{v}^*\mathbf{v}^{*\top}$ as the variance. For each v_i, with probability a_i, we let $v_i = 1$, with probability $1 - v_i$, we let $v_i = 0$.

Through the rounding technique, we can get a feasible $\mathbf{v}$ denoted by $\mathbf{v}^f$ for the Problem (4.4), however, the solution we get is not necessarily optimal. Denote the optimal $\mathbf{v}$ by $\mathbf{v}^{opt}$. Let $f(\mathbf{v}) = \mathbf{vPv} + 2\mathbf{q}^\top\mathbf{v} + c_1$, the connection between $\mathbf{v}^f$, $\mathbf{v}^{opt}$ and $\tilde{\mathbf{V}}^*$ is $f(\mathbf{v}^f) \leq f(\mathbf{v}^{opt}) \leq \langle \tilde{\mathbf{P}}, \tilde{\mathbf{V}}^* \rangle$. Though we do not know the $\mathbf{v}^{opt}$, we can infer that the result is good enough if $f(\mathbf{v}^f)$ is close to $\langle \tilde{\mathbf{P}}, \tilde{\mathbf{V}}^* \rangle$.

4.3.2 Algorithm

The (4.8) is solvable through the optimizers like Mosek, Cplex, Gurobi. Assume that we have solved this semidefinite programming, we use the Gaussian randomization procedure described in Section 4.3.1 to discretize the $\mathbf{v}^*$ and get the final exposure $\mathbf{y}$ we want.

Algorithm 2 `GaussianSDP`

input : k, $\mathbf{s}$, n, $\mathbf{L}$, t
output : The changed exposure vector $\mathbf{y}$
Use solver to construct Equation (4.8) and get $\mathbf{V}^*$, $\mathbf{v}^*$.
Initialize $\mathbf{y} \leftarrow \mathbf{s}$, $f = 0$, initialize $\mathbf{v}$. Form covariance matrix $\Sigma \leftarrow \mathbf{V}^* - \mathbf{v}^*\mathbf{v}^{*\top}$ **for** $i \leftarrow 1, \ldots, t$ **do**
    sample $\mathbf{v}_0 \sim \mathcal{N}(\mathbf{v}^*, \Sigma)$ **do**
        $\mathbf{v}' \leftarrow$ randomized_rounding($\mathbf{v}_0$)
    while $\mathbf{e}^\top\mathbf{v}' > k$;
    if $f < \mathbf{v}'\mathbf{P}\ \mathbf{v} + 2\mathbf{q}^\top$ **then**
        $\mathbf{v} \leftarrow \mathbf{v}'$ and $f \leftarrow \mathbf{v}'\mathbf{P}\ \mathbf{v} + 2\mathbf{q}^\top$
for $i \leftarrow 1, \ldots, n$ **do**
    if $v_i == 1$ **then**
        $y_i \leftarrow 1$
    if $v_{i+n} == 1$ **then**
        $y_i \leftarrow -1$
return $\mathbf{y}$

The randomized_rounding is a subrountine with:

$$\mathbb{P}(\text{randomized\_rounding}(\mathbf{x})[i] = 1) = \max\{\min\{1, x_i\}, 0\}.$$

4.4 Greedy Algorithm

Greedy algorithms are among the most intuitive approaches to solve the Knapsack Problems (KP), and they have an $\frac{1}{2}$-approximation ratio. To put it simply, the greedy algorithm maximizes the profit of each step, and always keep the weight bounded. The Quadratic Knapsack Problem (QKP) is different from KP in two ways, first, the matrix $\mathbf{P}$ can have negative items; second, the profits of each step can not be calculated in advance. Regarding the different ways to decide the order of items, there are two greedy algorithms discussed by Julstrom *et al.* [47], we employ the methods with some changes. In the Absolute Greedy Algorithm, the order is calculated in advance, while in the Relative Greedy Algorithm, the order is calculated before each step. Our design of the greedy algorithm is based on the Problem (4.4), the equivalent form of the original problem.

4.4.1 Absolute Greedy Algorithm

Observe that whenever $v_i = 1$, $P_{i,i}$ and $2q_i$ is counted in; however, the value of $P_{i,j}$ is decided together by x_i and x_j. Analyzing the off-diagonal elements of $P_{i,j}$ can be too complicated, heuristically, we can use the diagonal values of $\mathbf{P}$ and $\mathbf{q}$ to decide the order of choosing the items. As show in Algorithm 3, the absolute greedy algorithm is very quick. It takes $\mathcal{O}(n\log(n))$ to sort $diag(\mathbf{P}) + 2\mathbf{q}$, in the worst case, all the entries will be traveled, it takes $\mathcal{O}(kn)$.

Algorithm 3 `AbGreedy`

```
input  : L, k, n
output : The changed exposure vector y
Initialize y ← s, P, q; Sort [2n] according to P_{i,i} + 2q_i, i = 1, ..., n, in descending order;
  Store in L  Initiate a list S, a counter C = 0
for i in L do
    if P_{i,i} + 2 ∑_{j∈S} P_{i,j} + 2q_i > 0 then
        Add i to S; v_i ← 1; C ← C + 1;
        if C ≥ k then
            break
for i ← 1, ..., n do
    if v_i == 1 then
        y_i ← 1
    if v_{i+n} == 1 then
        y_i ← −1
return y
```

4.4.2 Relative Greedy Algorithm

In the Relative Greedy Algorithm, item to be picked is decided before each step. The time complexity of this algorithm is $\mathcal{O}(kn^2)$.

Algorithm 4 `RtGreedy`

---

input : $\mathbf{L}$, k, n
output : The changed exposure vector $\mathbf{y}$
Initialize $\mathbf{y} \leftarrow \mathbf{s}$, $\mathbf{P}$, $\mathbf{q}$; Store i with largest $P_{i,i} + 2q_i, i = 1, \ldots, n$ into L; Store indexes other than i into S;
while $|L| < k$ **do**
  for i *in* S **do**
    Find i with largest $P_{i,i} + 2\sum_{j\in L} P_{i,j} + 2q_i$
  if $P_{i,j} + 2\sum_{j\in L} P_{i,j} + 2q_i + \leq 0$ **then**
    Break
  Add i to L, delete i in S; $v_i \leftarrow 1$;
for $i \leftarrow 1, \ldots, n$ **do**
  if $v_i == 1$ **then**
    $y_i \leftarrow 1$
  if $v_{i+n} == 1$ **then**
    $y_i \leftarrow -1$
return $\mathbf{y}$

---

5 ℓ_2-bounded Diversity Maximization

In Chapter 4 we build a continuous diversity maximization model that is restricted to a box constraint. According to the analysis in Theorem 3.1, the model is reformulated as a Integer Programming problem, and the changed *exposure index*es always take the extreme values, respectively 1 and -1. However, changing the users exposure indexes to extreme values does not meet the request in real world situations based on the following intuitive assumptions that: First, the users do not want their information exposure to be dramatically changed, since they do not want to receive information that absolutely challenges their ideas; they will abandon the platform if the recommendation system does so. Second, if the users' exposures to the information are totally different than the value of their friends, disputes will arise in the platform since they receive totally different information and the communication will be harmed. To tackle this situation, we add another constraint to our model that can prevent the changed *exposure index*es from taking extreme values.

5.1 Problem Formulation

We use the same notation as in Chapter 4. We briefly list the relevant ones here. Let $G = (V, E, w)$ be a simple weighted graph that represents the network; let $w(i, j)$ be the connection strength of node i and j, let $\mathbf{L}$ be the unnormalized Laplacian matrix of G. Let $\mathbf{X}$ and $\mathbf{Y}$ be any $m \times n$ matrix, let $\langle \mathbf{X}, \mathbf{Y} \rangle = \sum_{i=1}^{n} \sum_{j=1}^{n} X_{ij} Y_{ij}$. Denote by $diag(\mathbf{X})$ the diagonal vector of $\mathbf{X}$. Denote by $Diag(\mathbf{x})$ a diagonal matrix whose diagonal equals to $\mathbf{x}$. Denote by $rank(\mathbf{X})$ the rank of $\mathbf{X}$. Let $\mathbf{x}$ be a vector, denote by $card(\mathbf{x})$ the number of non-zero elements in $\mathbf{x}$. Let $\mathbf{e}$ be a vector of all ones. Let $\mathbf{s}$ be a *exposure index* vector in G, then the *diversity index* of G is $\eta(G, \mathbf{s}) = \sum_{(i,j) \in E} w(i, j)(s_i - s_j)^2$.

The objection of our model is to maximize the *diversity index* by changing a limited number of exposure indexes; besides keeping the cardinality constraint and the box constraint, we want to avoid extreme values on $\mathbf{x}$. The ℓ_2 constraint on the total changes can help achieving our goal. In this way, the feasible area without the cardinality constraint is no longer a polygon, and the problem can not be reduced to searching through the extreme points when the cardinality constraint is added. Formally, we define the problem in Problem 5.1.

Problem 5.1 (ℓ_2-bounded diversity maximization)**.** *Given a graph $G = (V, E, w)$, a nonnegative constant α and an exposure index vector $\mathbf{s}$ with $s_i \in [-1, 1], i = 1, \ldots, n$; change at most k elements in $\mathbf{s}$ to get a new exposure indanotherex $\mathbf{y}$ with $y_i \in [-1, 1], i = 1, \ldots, n$, such that $\|\mathbf{y} - \mathbf{s}\|_2^2 \leq \alpha$, and the new diversity index $\eta(G, \mathbf{y})$ is maximized.*

We use the equivalent form of the *diversity index* of G given the *exposure index* $\mathbf{s}$ as $\eta(G, \mathbf{s}) = \mathbf{sLs}$, the problem has a direct formulation as in Equation (5.1). For the convenience of illustration, the constraints are annotated with prefix Con and a number.

$$
\begin{aligned}
\underset{\mathbf{y}}{\text{maximize}} \quad & \mathbf{y}^\top \mathbf{L} \mathbf{y} \\
\text{subject to} \quad & \text{Con1: } \|\mathbf{y}\|_\infty \le 1, \\
& \text{Con2: } \|\mathbf{y}-\mathbf{s}\|_2^2 \le \alpha, \\
& \text{Con3: } \mathit{card}(\mathbf{y}-\mathbf{s}) \le k.
\end{aligned} \tag{5.1}
$$

The Problem (5.1) is a maximization problem with three constraints. As we have discussed in the Chapter 3, there are papers that have solved the related problems. In conclusion, Boyd *et al.* [28, Appendix B] prove that when there is a single constraint **Con2**, the global maximum can be found by Semidefinite Relaxation; Daspremont *et al.* [35] give a semidefinite relaxation method to solve the problem with constraints **Con2** and **Con3**; while the problem with only constraints **Con1** and **Con3** have already been solved in previous chapter, Ye *et al.* [36] give a general formed quadratic programming with a box constraint.

5.2 A Two-step Method

In Chapter 4 we have solved Problem (5.1) with constraints **Con1** and **Con3**, heuristically, we can employ and adapt the previous results to fit the **Con2**.

Assume that Problem (4.1) is solved and we get the optimal solution $\mathbf{y}$, if it follows that $||\mathbf{y}-\mathbf{s}||_2^2 \le \alpha$, then the optimal solution of Problem (4.1) is the optimal solution of Problem (5.1), we are done; otherwise we can infer which elements in $\mathbf{s}$ have been changed by checking the non-zero elements in $\mathbf{y}-\mathbf{s}$. In this two-step heuristic method, we first restrict the changes on these elements, then determine the amount of changes.

Let $\mathbf{x} \in \mathbb{R}^k$ be a vector of all the non-zero elements in $\mathbf{y}-\mathbf{s}$ with x_j being the jth non-zero element in $\mathbf{y}-\mathbf{s}$. Let $\mathbf{M} \in \mathbb{R}^{n\times k}$ be a matrix with

$$
M_{ij} = \begin{cases} 1 & \text{if} \quad x_j \neq 0 \text{ and } y_i - s_i = x_j, \\ 0 & \text{if} \quad \text{otherwise.} \end{cases} \tag{5.2}
$$

Lemma 5.2. *Let vector* $\mathbf{x}$ *be an optimal solution of Problem* (5.3)*, then vector* $\mathbf{s}+\mathbf{M}\mathbf{x}$ *is a feasible solution of Problem* (5.1)*.*

$$
\begin{aligned}
\underset{\mathbf{x}}{\textit{maximize}} \quad & (\mathbf{s}+\mathbf{M}\mathbf{x})^\top \mathbf{L}(\mathbf{s}+\mathbf{M}\mathbf{x}) \\
\textit{subject to} \quad & \|\mathbf{s}+\mathbf{M}\mathbf{x}\|_\infty \le 1, \\
& \mathbf{x}^\top \mathbf{x} \le \alpha.
\end{aligned} \tag{5.3}
$$

Proof. By the form of the $\mathbf{M}$, the cardinality constraint, Con3 is naturally satisfied. Con1 and Con2 correspond to $\|\mathbf{s}+\mathbf{M}\mathbf{x}\|_\infty \le 1$ and $\mathbf{x}^\top\mathbf{x} \le \alpha$ respectively. □

The Problem (5.3) is a convex maximization problem restricted to the ℓ_∞ and ℓ_2 constraint. In the Lemma 5.3 we show Problem (5.3) can be solved through semidefinite relaxation.

We introduce a variable $\mathbf{X} \in \mathbb{R}^{k\times k}$. Let

$$\tilde{\mathbf{P}} = \begin{pmatrix} \mathbf{M}^\top \mathbf{L}\mathbf{M} & \mathbf{M}^\top \mathbf{L}\mathbf{s} \\ \mathbf{s}^\top \mathbf{L}\mathbf{M} & \mathbf{s}^\top \mathbf{L}\mathbf{s} \end{pmatrix},$$

and

$$\tilde{\mathbf{X}} = \begin{pmatrix} \mathbf{X} & \mathbf{x} \\ \mathbf{x}^\top & 1 \end{pmatrix}.$$

Lemma 5.3. *Problem (5.4) is a relaxation of Problem (5.3).*

$$\begin{aligned} \underset{\mathbf{x},\mathbf{X}}{\textit{maximize}} \quad & \langle \tilde{\mathbf{X}}, \tilde{\mathbf{P}} \rangle \\ \textit{subject to} \quad & \tilde{\mathbf{X}} = \begin{pmatrix} \mathbf{X} & \mathbf{x} \\ \mathbf{x}^\top & 1 \end{pmatrix}, \\ & \tilde{\mathbf{X}} \succcurlyeq 0, \\ & \|\mathbf{s} + \mathbf{M}\mathbf{x}\|_\infty \le 1, \\ & \langle \mathbf{X}, \mathbf{I} \rangle \le \alpha, \\ & \|\mathbf{s}\mathbf{s}^\top + \mathbf{M}\mathbf{x}\mathbf{s}^\top + \mathbf{s}\mathbf{x}^\top\mathbf{M}^\top + \mathbf{M}\mathbf{X}\mathbf{M}^\top\|_\infty \le 1. \end{aligned} \tag{5.4}$$

Proof. Adding a condition that the rank of $\tilde{\mathbf{X}}$ equals to 1 to the Problem (5.4), then two problems are equivalent. The condition $||\mathbf{s}\mathbf{s}^\top + \mathbf{M}\mathbf{x}\mathbf{s}^\top + \mathbf{s}\mathbf{x}^\top\mathbf{M}^\top + \mathbf{M}\mathbf{X}\mathbf{M}^\top||_\infty \le 1$ is equivalent with $||\mathbf{s} + \mathbf{M}\mathbf{x}||_\infty \le 1$. Problem (5.4) is a relaxed form of Problem (5.3) since the rank constraint is relaxed. □

5.2.1 Rounding Technique

Assume that we have solved the problem and let $\mathbf{x}^*$ and $\mathbf{X}^*$ be the optimal variables. According to the condition $\langle \mathbf{X}^*, \mathbf{I} \rangle \le \alpha$ and $\tilde{\mathbf{X}}^* \succcurlyeq 0$, we can infer that $\mathbf{x}^{\top *}\mathbf{x}^* \le \alpha$, thus $\mathbf{x}^*$ is a feasible solution of Problem (5.3). We do not use $\mathbf{x}^*$ as the solution of Problem (5.3), but we draw a $\mathbf{x}' \sim \mathcal{N}(\mathbf{x}^*, \mathbf{X}^* - \mathbf{x}^*\mathbf{x}^{*\top})$ that can be a better solution than $\mathbf{x}^*$, we illustrate the algorithm in Section 5.2.2.

5.2.2 Algorithm

The `TwoStepSDP` involves solving two semidefinite programming problems, Problem (4.8) and Problem (5.4) respectively. The complexity of the latter one is partially based on the size of the variables, which are $k \times k$ for $\mathbf{X}$ and k for $\mathbf{x}$. When k is significantly smaller than n, for instance, $k = 0.1n$, solving the latter one will take smaller amount of time.

There are two special cases for solving Problem (5.4), the first one is that after implementing Algorithm 2, the condition $\|\mathbf{y} - \mathbf{s}\|_2^2 \le \alpha$ is satisfied, then $\mathbf{y}$ is already the solution; the second one is when $k = n$, there is no need to implement Algorithm 2.

The randomized_rounding is a subrountine with:

$$\mathbb{P}(\text{randomized\_rounding}(\mathbf{x})[i] = 1) = \max\{\min\{1, x_i\}, 0\}.$$

Algorithm 5 `TwoStepSDP`

input : k, $\mathbf{s}$, α, n, $\mathbf{L}$, t
output : The changed exposure vector $\mathbf{y}$
Initialize $\mathbf{y} \leftarrow \mathbf{s}$;
if $k \neq n$ **then**
  Implement Algorithm 2, get solution $\mathbf{y}_0$;
  if $\|\mathbf{y}_0 - \mathbf{s}\|_2^2 \leq \alpha$ **then**
    $\mathbf{y} \leftarrow \mathbf{y}_0$; **return y**
Construct $\mathbf{M}$ according to Equation (5.2);
Use solver to get results of Equation (5.4) $\mathbf{x}^*$ and $\mathbf{X}^*$; $\Sigma = \mathbf{X}^* - \mathbf{x}^*\mathbf{x}^{*\top}$;
for $i \leftarrow 1, \ldots, t$ **do**
  while $\|\mathbf{x}'\|_2^2 > \alpha$ **do**
    sample $\mathbf{x}' \sim \mathcal{N}(\mathbf{x}^*, \Sigma)$;
    $\mathbf{x}' \leftarrow$ randomized_rounding$(\mathbf{x})$;
  if $f < \mathbf{x}'^\top\mathbf{M}^\top\mathbf{L}\mathbf{M}\mathbf{x}' + 2\mathbf{x}'^\top\mathbf{M}^\top\mathbf{L}\mathbf{s}$ **then**
    $\mathbf{x} \leftarrow \mathbf{x}'$ and $f \leftarrow \mathbf{x}'^\top\mathbf{M}^\top\mathbf{L}\mathbf{M}\mathbf{x}' + 2\mathbf{x}'^\top\mathbf{M}^\top\mathbf{L}\mathbf{s}$
$\mathbf{y} \leftarrow \mathbf{s} + \mathbf{M}\mathbf{x}$;
return y

5.3 Semidefinite Relaxation

Instead of solving Problem (5.1) in a two-step manner, we can directly apply Semidefinite Relaxation to solve this problem. To be specific, we solve Problem (5.1) by solving a relaxed Problem (5.6), then rounding the solutions to ensure they are feasible for the original problem.

Lemma 5.4. *Problem (5.5) is an equivalent form of Problem (5.1) with respect to the objective function.*

$$\begin{aligned} \underset{\mathbf{x}}{\textit{maximize}} \quad & \left\langle \begin{bmatrix} \mathbf{X} & \mathbf{x} \\ \mathbf{x}^\top & 1 \end{bmatrix}, \begin{bmatrix} \mathbf{L} & \mathbf{L}\mathbf{s} \\ \mathbf{s}^\top\mathbf{L} & \mathbf{s}^\top\mathbf{L}\mathbf{s} \end{bmatrix} \right\rangle \\ \textit{subject to} \quad & \|\mathbf{x} + \mathbf{s}\|_\infty \leq 1 \\ & \begin{bmatrix} \mathbf{X} & \mathbf{x} \\ \mathbf{x}^\top & 1 \end{bmatrix} \succcurlyeq 0 \\ & \textit{rank}\left(\begin{bmatrix} \mathbf{X} & \mathbf{x} \\ \mathbf{x}^\top & 1 \end{bmatrix} \right) = 1 \\ & \langle \mathbf{X}, \mathbf{I} \rangle \leq \alpha \\ & \textit{card}(\mathbf{x}) \leq k \end{aligned} \tag{5.5}$$

Proof. Let $\mathbf{x} = \mathbf{y} - \mathbf{s}$, Problem (5.1) is equivalent with Problem (5.5) with respect to objective function and the constraints. □

Lemma 5.5 (Daspremont *et al.* [35]). *The condition $card(\mathbf{x}) \leq k$ can be relaxed into $\mathbf{e}^\top|\mathbf{X}|\mathbf{e} \leq \alpha k$.*

Proof. According to the Cauchy–Schwarz inequality, $\|\mathbf{x}\|_1^2 \leq card(\mathbf{x})\|\mathbf{x}\|_2^2$. Thus $\|\mathbf{x}\|_1^2 = \mathbf{e}^\top|\mathbf{x}||\mathbf{x}|^\top\mathbf{e} = \mathbf{e}^\top|\mathbf{X}|\mathbf{e}$, $\|\mathbf{x}\|_2^2 = \langle \mathbf{X}, \mathbf{I}\rangle$, and $card(\mathbf{x}) \leq k$ imply that $\mathbf{e}^\top|\mathbf{X}|\mathbf{e} \leq k\alpha$. The $card(\mathbf{x})$ can be relaxed into a looser condition $\mathbf{e}^\top|\mathbf{X}|\mathbf{e} \leq \alpha k$. □

Comparing with Section 3.2.3, the Problem (5.5) has one extra constraint: $||\mathbf{x} + \mathbf{s}||_\infty \leq 1$, since we know that the infinity norm ball is a convex set, the constraint can be kept as it is. Besides, the constraint implies that $X_{i,i} \leq s_i^2 + 2|s_i| + 1, i = 1, \ldots, n$. In conclusion, we can write the programming as Problem (5.6).

Lemma 5.6. *Problem (5.6) is a relaxation of Problem (5.5).*

$$\begin{aligned}
\underset{\mathbf{x},\mathbf{X}}{\textit{maximize}} \quad & \left\langle \begin{bmatrix} \mathbf{X} & \mathbf{x} \\ \mathbf{x}^\top & 1 \end{bmatrix}, \begin{bmatrix} \mathbf{L} & \mathbf{L}\mathbf{s} \\ \mathbf{s}^\top\mathbf{L} & \mathbf{s}^\top\mathbf{L}\mathbf{s} \end{bmatrix} \right\rangle \\
\textit{subject to} \quad & \begin{bmatrix} \mathbf{X} & \mathbf{x} \\ \mathbf{x}^\top & 1 \end{bmatrix} \succcurlyeq 0 \\
& \mathbf{e}^\top|\mathbf{X}|\mathbf{e} \leq k\alpha \\
& \langle \mathbf{X}, \mathbf{I} \rangle \leq \alpha \\
& \|\mathbf{x} + \mathbf{s}\|_\infty \leq 1 \\
& X_{i,i} \leq s_i^2 + 2|s_i| + 1, i = 1, \ldots, n
\end{aligned} \tag{5.6}$$

Proof. According to Lemma 5.5, for any $\mathbf{x}, \mathbf{X}$ that satisfy $card(\mathbf{x}) \leq k$ and $\mathbf{X} = \mathbf{x}\mathbf{x}^\top$, it follows that $\langle \mathbf{X}, \mathbf{I}\rangle \leq k\alpha$. For any $\mathbf{x}, \mathbf{X}$ that satisfy $\|\mathbf{x} + \mathbf{s}\|_\infty \leq \alpha$ and $\mathbf{X} = \mathbf{x}\mathbf{x}^\top$, it follows that $X_{i,i} \leq s_i^2 + 2|s_i| + 1, i = 1, \ldots, n$. As a result, any feasible solution $\mathbf{x}, \mathbf{X}$ of Problem (5.5), is also a feasible solution of Problem (5.6). Thus the lemma holds. □

5.3.1 Rounding Technique

Assume that we have got the optimal solutions $\mathbf{x}^*$ and $\mathbf{X}^*$ of Problem (5.6). In this problem setting, the discretization is no longer needed. Heuristically, we draw a sample $\mathbf{x}' \sim \mathcal{N}(\mathbf{x}^*, \mathbf{X}^* - \mathbf{x}^*\mathbf{x}^{*\top})$, and keep the top k elements of $\mathbf{x}'$ regarding $|\mathbf{x}'|$ such that the ℓ_2 constraint is satisfied.

5.3.2 Algorithm

Solving Problem (5.6) by the solver, we can get $\mathbf{y}$ through the Algorithm 6. In the algorithm, top$(\mathbf{x}_0, k)$ is a subroutine that returns a $\mathbf{x}'$ such that:

$$x'_i = \begin{cases} \max\{\min\{x_{0i}, 1\}, 0\} & \text{if} \quad |x_{0i}| \text{ ranks top } k \text{ in list } |x_0|, \\ 0 & \text{if} \quad \text{otherwise.} \end{cases}$$

.

Algorithm 6 `GaussianSDPtop`

input : k, $\mathbf{s}$, n, α, $\mathbf{L}$
output : The changed exposure vector $\mathbf{y}$
Use solver to construct Equation (5.6) and get $\mathbf{X}^*$, $\mathbf{x}^*$.
Initialize $\mathbf{y} \leftarrow \mathbf{s}$, $f = 0$, initialize $\mathbf{x}$. Form covariance matrix $\Sigma \leftarrow \mathbf{X}^* - \mathbf{x}^*\mathbf{x}^{*\top}$
for $i \leftarrow 1, \ldots, 1000$ **do**
  sample $x_0 \sim \mathcal{N}(\mathbf{x}^*, \Sigma)$ **do**
    $\mathbf{x}' \leftarrow \mathrm{top}(\mathbf{x}_0, k)$
  while $\mathbf{x}'^\top \mathbf{x}' > \alpha$;
  if $f < \mathbf{x}'\mathbf{L}\ \mathbf{x}' + 2\mathbf{x}'^\top \mathbf{L}\mathbf{s}$ **then**
    $\mathbf{x} \leftarrow \mathbf{x}'$ and $f \leftarrow \mathbf{x}'\mathbf{L}\ \mathbf{x}' + 2\mathbf{x}'^\top \mathbf{L}\mathbf{s}$
$\mathbf{y} \leftarrow \mathbf{s} + \mathbf{x}$; **return** $\mathbf{y}$

5.4 Greedy Algorithm

Similarly as Chapter 4, we can use the greedy algorithm to solve the problem. The algorithms we apply to solve the Problem 5.1 is similar with Algorithm 3 and Algorithm 4; the different part is that the condition $\mathbf{x}^\top \mathbf{x}$ will be checked at every step.

5.4.1 Absolute Greedy Algorithm

Algorithm 7 `AbGreedy`

input : $\mathbf{L}$, k, n
output : The changed exposure vector $\mathbf{y}$
Initialize $\mathbf{y} \leftarrow \mathbf{s}$, $\mathbf{P}$, $\mathbf{q}$; Sort $[2n]$ according to $P_{i,i} + 2q_i, i = 1, \ldots, n$ in descending order and store in list L; Initiate a list S, a counter $C = 0$ and an accumulator $T = 0$
for i *in* L **do**
  if $P_{i,i} + 2\sum_{j \in S} P_{i,j} + 2q_i > 0$ **then**
    if $i <= n$ **then**
      $T \leftarrow T + (1 - s_i)^2$
    else
      $T \leftarrow T + (-1 - s_{i-n})^2$
    Add i to S; $v_i \leftarrow 1$; $C \leftarrow C + 1$; **if** $C \geq k$ *or* $T > \alpha$ **then**
      break
for $i \leftarrow 1, \ldots, n$ **do**
  if $v_i == 1$ **then**
    $y_i \leftarrow 1$
  if $v_{i+n} == 1$ **then**
    $y_i \leftarrow -1$
return $\mathbf{y}$;

5.4.2 Relative Greedy Algorithm

Algorithm 8 `RtGreedy`

input : $\mathbf{L}$, k, n

output : The changed exposure vector $\mathbf{y}$

Initialize $\mathbf{y} \leftarrow \mathbf{s}$, $\mathbf{P}$, $\mathbf{q}$; Store i with largest $P_{i,i} + 2q_i, i = 1, \ldots, n$ into set L; Store indexes other than i into set S, initialize an accumulator $T = 0$

while $|L| < k$ **do**

  for i *in* S **do**

    Find i with largest $P_{i,i} + 2\sum_{j \in L} P_{i,j} + 2q_i$

  if $i <= n$ **then**

    $T \leftarrow T + (1 - s_i)^2$

  else

    $T \leftarrow T + (-1 - s_{i-n})^2$

  if $P_{i,j} + 2\sum_{j \in L} P_{i,j} + 2q_i + \leq 0$ *or* $T > \alpha$ **then**

    Break

  Add i to L, delete i in S; $v_i \leftarrow 1$;

for $i \leftarrow 1, \ldots, n$ **do**

  if $v_i == 1$ **then**

    $y_i \leftarrow 1$

  if $v_{i+n} == 1$ **then**

    $y_i \leftarrow -1$

return $\mathbf{y}$;

6 Experiments and Results

In this section, we give a comprehensive description on the dataset that we have used for the experiment, and the detailed results of the experiments.

6.1 Data Set Description

Zachary's Karate Club [48] is a social network of a karate club that shows the connections between the members inside the group. It captures the situation when conflicts arose in the club and the members are split into two groups.

Dolphin [49] is a communication network of a group of 62 dolphins. Like the Zachary's Katate Club, there are two separated communities.

Polblogs [8] is directed network of the hyperlinks between blogs on US politics. The network was collected in 2005. Each blog was assigned with an attribute that indicated its political leaning, left or right.

Polbooks [50] is network of books about US politics sold by the Amazon.com. It was collected around the 2004 presidential election. Each book was assigned with an attribute that indicated its political leaning, left, right or neutral. The edge between two books indicates a frequent co-purchasing among them.

Twitter [12] is a network collected among Twitter users between 2011 and 2016. Based on the tendencies of the controversial issues, such as *gun control*, *abortion* and *ObamaCare* shown on their time line, the users were grouped into two communities.

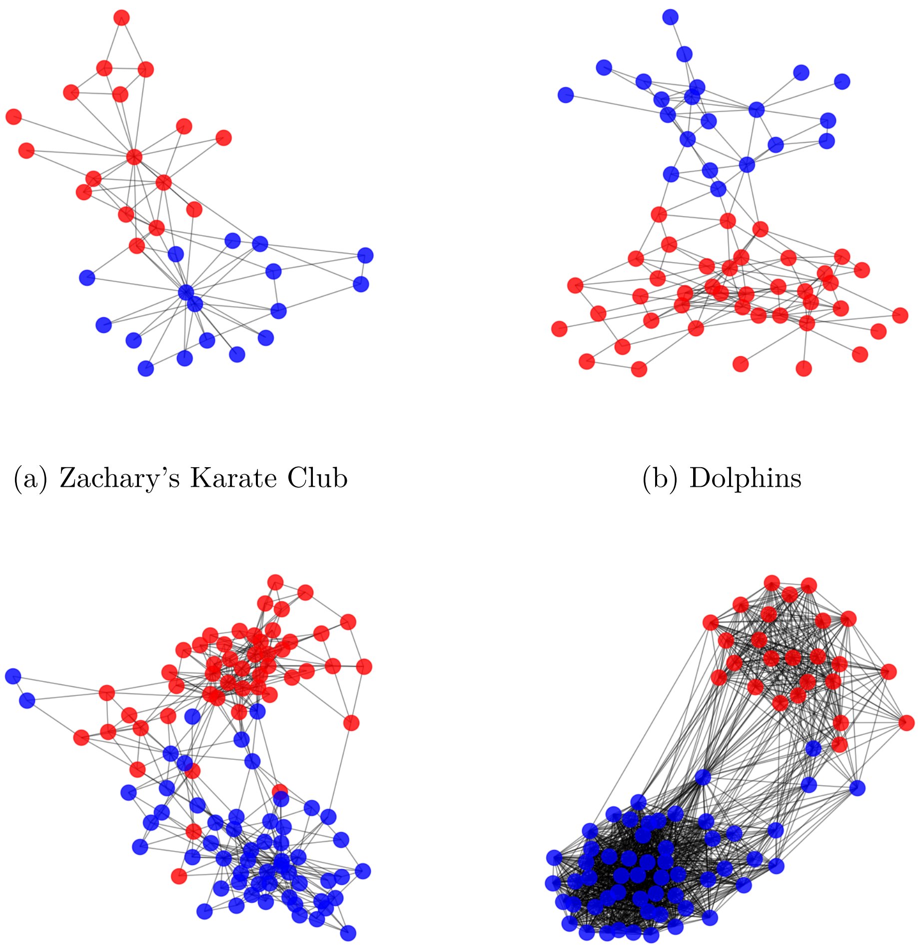
We can list the parameters of the datasets in Table 1. The Positive and Negative in Tabel 1 mean the number of nodes in the two communities of the network, the names are inter-changeable.

Table 1: Data Sets

Dataset	Nodes	Edges	Positive	Negative
Karate	34	78	17	17
Dolphin	62	159	41	21
Twitter	80	1403	25	55
Polbooks	105	441	43	49
Polblogs	1222	16717	636	586

We visualize the four social networks in Figure 2 to obtain a intuitive sense of polarization. The *Polbooks* and *twitter* are particularly political related, and the labels were collected. These two graphs, together with the labels on the nodes show the political polarization. There are also obvious two communities in the *Zachary's Karate Club* and the *Dolphins*, however, the labels are not politically related.

Apart from the the already existing labels, as a comparison, we randomly assign labels to the nodes in the graph with a probability as the proportion of the labels in nodes. We annotate the datasets with randomly labeled nodes by appending "_R" after their names, for instance, Karate is the original dataset and Karate_R is the



(a) Zachary's Karate Club

(b) Dolphins

(c) Polbooks

(d) Twitter

Figure 2: Polarized Social Networks

dataset with randomly labeled nodes. We can get a table of parameters of all the datasets in Table 2. The experiment is implemented on the weighted graphs, we randomly assign each edge with weight in $[0, 1]$, the *exposure index* of each node is initialized by -1 or 1 multiplying a value in $[0, 1]$. -1 and 1 are dependent on their labels.

6.2 Setup

The experiments are run on one node of Aalto Triton, which is a part of the Finnish Grid and Cloud Infrastructure, the CPU setting is 2x12 core Xeon E5 2680 v3 2.50GHz and the Memory setting is 50 G. The optimization solvers include the Mosek Fusion and the Cplex. The Mosek Fusion is applied to model and solve the SDP problems and the Cplex is applied to solve the IQP problems.

Table 2: Datasets

Dataset	Nodes	Edges	Degree	Positive	Negative
Karate	34	78	4.59	17	17
Karate_R	34	78	4.59	18	16
Dolphine	62	159	5.13	41	21
Dolphine_R	62	159	5.13	43	19
Twitter	80	1403	35.08	25	55
Twitter_R	80	1403	35.08	27	53
Polbooks	105	441	8.40	49	56
Polbooks_R	105	441	8.40	50	55
Polblogs	1224	16717	27.31	732	758
Polblogs_R	1224	16717	27.31	710	780

6.3 Bounded-box Diversity Maximization

We list the final results of the Problem 4.1 regarding the *diversity index* in Table 3 and the running time in Table 4. For the head of the table, the k denotes the cardinality constraint on the changes of the *exposure index*, respectively, we take 10%, 20% and 100% of the order n of the graph. The `IQP` refers solving the problem directly as a mixed-integer problem; its results are the optimal solutions. `SDPMax` refers to the direct results we got by solving Problem (4.8), these values are the upper bounds for the real solutions. The `SimpleSDP` and `GaussianSDP` refer to the results obtained through two rounding techniques after solving Problem (4.8), the former rounds the variables directly, while the latter refers to Algorithm 2. `AbGreedy` refers the Absolute greedy Algorithm 3 and the `RtGreedy` refers the relative greedy Algorithm 4.

6.3.1 Evaluation

The result of the `IQP` is the optimal solution we get for the problem. Besides `IQP`, out of 24 cases (from the karate to twitter), the `GaussianSDP` has 8 cases that reach the best results, while the `RtGreedy` has 18 cases that reach the best results. To our surprise, the `RtGreedy` is even better than the `GaussianSDP`. When k is large then out of 8 cases, the `GaussianSDP` has 5 cases that reach the best results, and the `RtGreedy` has 3 cases that reach the best results, so the `GaussianSDP` is better when k is large. Actually when k is set as the same of n, then the problem is degenerated into a max cut problem 3.4.1 and the solution is guaranteed to be in expectation 0.87 of the optimal solution.

When compared with `GaussianSDP` and the `RtGreedy`, the results of the `AbGreedy` are worse in almost all the cases, while the results of `SimpleSDP` are comparable with these two methods when cardinality bound k is small.

Table 3: Result of bounded-box diversity maximization

Dataset(Diversity Index)	k	IQP	SimpleSDP	GaussianSDP	SDPMax	AbGreedy	RtGreedy
Karate (6.94)	$0.1n$	50.77	50.77	50.77	57.08	48.73	50.77
	$0.2n$	72.00	70.77	69.17	84.44	68.59	70.77
	n	136.06	105.74	135.84	138.53	114.83	121.90
Karate_R (21.88)	$0.1n$	43.85	40.66	43.85	54.59	43.85	43.85
	$0.2n$	58.89	54.12	54.51	75.08	56.24	58.30
	n	126.52	105.52	126.52	129.23	108.22	111.60
Dolphine (11.95)	$0.1n$	79.71	73.77	73.39	99.67	76.54	79.07
	$0.2n$	130.56	107.08	110.87	155.80	100.14	117.08
	n	275.05	226.54	273.27	281.74	221.47	262.42
Dolphine_R (44.49)	$0.1n$	89.84	81.01	82.37	105.79	77.47	89.84
	$0.2n$	126.33	112.91	119.06	147.20	97.64	121.52
	n	232.99	186.66	226.34	239.88	190.25	227.89
Polbooks (49.99)	$0.1n$	256.42	238.14	249.91	280.68	253.58	256.42
	$0.2n$	361.93	328.23	326.64	416.30	312.32	360.00
	n	684.35	521.69	663.57	704.44	619.12	658.01
Polbooks_R (129.04)	$0.1n$	245.07	219.73	231.36	287.20	226.38	245.07
	$0.2n$	334.42	278.77	299.28	394.15	296.84	330.40
	n	624.30	475.00	609.54	647.57	522.49	612.66
Twitter (131.41)	$0.1n$	533.83	512.24	510.48	592.97	514.74	526.09
	$0.2n$	–	745.31	742.16	918.13	722.13	788.33
	n	–	1537.87	1689.95	1747.43	1525.38	1649.64
Twitter_R (427.92)	$0.1n$	680.68	660.87	649.37	767.06	656.62	678.88
	$0.2n$	–	774.80	809.02	1011.31	762.94	868.20
	n	–	1492.74	1695.75	1742.29	1424.36	1669.06
Polblogs (2303.47)	$0.1n$	–	12035.21	–	15069.29	12225.70	13541.96
	$0.2n$	–	14029.84	–	19207.44	13786.92	17151.97
Polblogs_R (5488.53)	$0.1n$	–	10615.77	–	15055.84	11527.70	12437.92

6.3.2 Scalability

We annotate – on the tables when the algorithm runs more than 1800 seconds. We can see from the results that the greedy algorithms have good performance on scaling into large datasets, at least it can handle the graphs with over one hundreds nodes. One the contrary, the `IQP` will run out of time for graphs of over hundreds of nodes or the dense graphs over tens of nodes (twitter). The time taken by Semidefinite relaxation without rounding (`SDPMax`) is comparable with that of the greedy algorithms on graphs with less than one hundred nodes. However, when the graph has more than one hundred nodes, the time consuming explodes, and the Semidefinite relaxation is no longer suitable. When comparing the time spent by the `GaussianSDP`, the `SimpleSDP` and the `SDPMax`, we find out that the Gaussian randomize procedure takes too much time to round the results.

Table 4: Running Time(s) of of bounded-box diversity maximization

Dataset	k	IQP	SimpleSDP	GaussianSDP	SDPMax	AbGreedy	RtGreedy
Karate	$0.1n$	1.43	1.03	1.45	0.55	0.00	0.00
	$0.2n$	69.43	1.00	1.53	0.55	0.00	0.00
	n	125.95	1.08	1.32	0.56	0.00	0.01
Karate_R	$0.1n$	51.73	1.04	1.49	0.55	0.00	0.00
	$0.2n$	119.76	1.03	1.65	0.55	0.00	0.00
	n	139.48	1.09	1.32	0.56	0.00	0.01
Dolphine	$0.1n$	104.63	2.79	26.21	1.83	0.00	0.00
	$0.2n$	134.50	2.80	27.89	1.82	0.00	0.01
	n	157.96	2.82	13.31	1.86	0.00	0.05
Dolphine_R	$0.1n$	0.69	2.74	24.24	1.82	0.00	0.00
	$0.2n$	235.53	2.82	24.91	1.84	0.00	0.01
	n	144.28	2.79	12.48	1.83	0.00	0.05
Polbooks	$0.1n$	102.93	7.74	402.80	5.46	0.17	0.03
	$0.2n$	181.86	7.83	281.01	5.44	0.47	0.05
	$n05$	417.69	7.98	142.02	5.50	3.37	0.23
Polbooks_R	$0.1n$	179.82	7.53	336.28	5.23	0.22	0.03
	$0.2n$	291.87	7.98	313.13	5.26	0.53	0.05
	n	1079.58	7.95	143.40	5.50	3.03	0.23
Twitter	$0.1n$	104.35	4.36	103.90	3.07	0.00	0.01
	$0.2n$	–	4.48	115.40	3.07	0.00	0.02
	n	–	4.35	44.52	3.10	0.00	0.10
Twitter_R	$0.1n$	492.08	4.12	84.78	3.03	0.00	0.01
	$0.2n$	–	4.39	91.68	3.00	0.00	0.01
	n	–	3.70	44.01	2.61	0.00	0.08
Polblogs	$0.1n$	–	2747.98	–	2722.72	4.68	16.50
	$0.2n$	–	2833.76	–	2808.58	21.39	58.58
Polblogs_R	$0.1n$	–	2447.00	–	2427.07	6.01	13.66

6.4 ℓ_2-bounded Diversity Maximization

We list the final results of Problem 5.1 regarding the *diversity index* in Table 5 and the running time in Table 6. For the head of the table, the k denotes the cardinality constraint on the changes of the *exposure index*, respectively, we take 10%, 20%, 30% and 100% of the order n of the graph. The vector $\mathbf{x}$ at the left side of each algorithm indicates the changes on the exposure vector $\mathbf{s}$ of that algorithm, and the parameter α is the upper bound on $||\mathbf{x}||_2^2$, i.e., $||\mathbf{x}||_2^2 \leq \alpha$. For our experiments we set $\alpha = \frac{1}{20}\alpha_{\max}$, where $\alpha_{\max} = \sum_i^n \max\{(1-\mathbf{s}_i)^2, (-1-\mathbf{s}_i)^2\}$ is the maximum value that $||\mathbf{x}||_2^2$ can take, taking into account the bounded-box constraints. We list the othter results by setting $\alpha = 0.1\alpha_{\max}$ and $\alpha = 0.2\alpha_{\max}$ at the Appendix A.

`SDPMax` refers to the direct results we got by solving Problem (5.6), these values are the upper bounds for the real solutions. The `GaussianSDPtop` rounds the results according to Algorithm 6. `TwoStepSDP` refers to the results got from a two-step sdp method 5. As a comparison, we adapted the `AbGreedy` and the `RtGreedy` in Problem 4.1 such that for each element in $\mathbf{s}$ that changed to 1 or -1, we check the total changes on $\mathbf{s}$ does not violate the ℓ_2 constraint.

6.4.1 Evaluation

When comparing the semidefinite relaxation based alogrihtms, the `TwoStepSDP` usually performs better than the `GaussianSDPtop`, except for 7 out of 30 cases; by comparing the relaxation value `SDPMax` with `TwoStepSDP`, we can see the the `TwoStepSDP` algorithm gives solutions of high quality, especially when $k = n$.

When k is assigned to $0.1n$, the `RtGreedy` is relatively better than all the other algorithms, however, it turns out the with the adapted gready algorithms `RtGreedy` and `AbGreedy` respectively, the process stops after a small number of elements in $\mathbf{x}$ are changed. Due to the ℓ_2 constraint no more elements in $\mathbf{x}$ can be changed, and as a result the value of the solution does not increase with k. We conclude that for Problem 5.1 the `RtGreedy` algorithm is not as effective as the `GaussianSDPtop` and `TwoStepSDP` algorithms.

Table 5: Result of ℓ_2-bounded diversity maximization

Dataset(Diversity Index)	k	α	$\|\|\mathbf{x}\|\|_2^2$	`GaussianSDPtop`	`SDPMax`	$\|\|\mathbf{x}\|\|_2^2$	`TwoStepSDP`	$\|\|\mathbf{x}\|\|_2^2$	`AbGreedy`	$\|\|\mathbf{x}\|\|_2^2$	`RtGreedy`
Karate (12.64)	$0.1n$	3.82	3.33	58.21	87.47	3.80	59.71	2.88	72.52	2.88	72.52
	$0.2n$	3.82	3.77	66.62	88.74	3.73	63.91	2.88	72.52	2.88	72.52
	$0.3n$	3.82	3.34	56.87	88.87	3.77	69.15	2.88	72.52	2.88	72.52
	n	3.82	3.18	57.97	88.87	3.82	74.65	2.88	72.52	2.88	72.52
Karate_R (44.23)	$0.1n$	3.82	3.28	83.28	139.59	1.45	84.05	1.89	82.42	2.11	84.69
	$0.2n$	3.82	1.83	104.74	146.65	3.66	92.99	1.89	82.42	3.04	115.41
	$0.3n$	3.82	3.12	114.54	149.61	3.70	110.82	1.89	82.42	3.04	115.41
	n	3.82	3.70	128.32	149.97	3.82	144.45	1.89	82.42	3.04	115.41
Dolphine (21.96)	$0.1n$	7.22	6.40	96.57	129.69	6.78	83.15	4.66	87.82	5.98	93.82
	$0.2n$	7.22	7.21	115.09	134.07	7.20	110.91	4.66	87.82	5.98	93.82
	$0.3n$	7.22	7.21	113.84	134.41	7.22	119.26	4.66	87.82	5.98	93.82
	n	7.22	6.91	110.70	134.41	7.16	118.50	4.66	87.82	5.98	93.82
Dolphine_R (90.44)	$0.1n$	7.22	3.84	144.12	233.17	6.94	155.01	5.89	141.73	7.22	163.75
	$0.2n$	7.22	4.66	167.56	243.98	7.22	179.67	5.89	141.73	7.22	163.75
	$0.3n$	7.22	5.24	182.73	248.42	7.22	189.19	5.89	141.73	7.22	163.75
	n	7.22	4.58	213.03	249.93	7.22	244.50	5.89	141.73	7.22	163.75
Twitter (247.72)	$0.1n$	9.25	5.28	615.36	969.13	9.24	790.18	6.89	661.92	8.13	754.89
	$0.2n$	9.25	7.25	731.64	1013.15	9.25	835.65	6.89	661.92	8.13	754.89
	$0.3n$	9.25	8.24	813.64	1036.86	9.25	862.28	6.89	661.92	8.13	754.89
	n	9.25	9.25	1055.41	1055.41	9.25	1055.41	6.89	661.92	8.13	754.89
Twitter_R (842.03)	$0.1n$	9.25	4.28	1212.13	1789.66	9.23	1222.83	7.46	1157.66	8.26	1302.78
	$0.2n$	9.25	4.89	1370.44	1903.76	9.09	1497.52	7.46	1157.66	8.26	1302.78
	$0.3n$	9.25	5.28	1546.35	1974.15	9.25	1629.27	7.46	1157.66	8.26	1302.78
	n	9.25	9.25	2084.27	2084.27	9.25	2084.27	7.46	1157.66	8.26	1302.78
Polbooks (98.30)	$0.1n$	12.04	9.09	326.63	477.71	11.75	351.47	11.16	359.88	11.91	427.80
	$0.2n$	12.04	8.88	337.57	486.15	12.04	407.45	11.16	359.88	11.91	427.80
	$0.3n$	12.04	9.83	360.31	486.27	12.04	409.98	11.16	359.88	11.91	427.80
	n	12.04	10.42	385.46	486.27	-	-	11.16	359.88	11.91	427.80
Polbooks_R (254.57)	$0.1n$	12.04	5.45	422.00	715.52	10.04	439.90	11.16	433.09	11.58	415.40
	$0.2n$	12.04	6.39	517.22	746.86	10.33	467.32	11.16	433.09	11.58	415.40
	$0.3n$	12.04	6.49	586.56	757.94	10.15	561.25	11.16	433.09	11.58	415.40
	n	12.04	8.25	661.19	761.31	-	-	11.16	433.09	11.58	415.40

6.4.2 Scalability

When comparing the time of `GaussianSDPtop` and `SDPMax`, we can see it takes little time to round the results. Comparing the time of `GaussianSDPtop` and `TwoStepSDP`, when $k = 0.1n$ or $k = 0.2n$, it takes smaller amount of time to solve `TwoStepSDP`, however, the time complexity increases significantly with increasing k. For larger datasets, the `GaussianSDPtop` becomes less efficient even with small k, while it is still possible to use `TwoStepSDP`. The greedy algorithm is more scalable.

Table 6: Running Time(s) of ℓ_2-bounded diversity maximization

Dataset	k	`GaussianSDPtop`	`SDPMax`	`AbGreedy`	`RtGreedy`	`TwoStepSDP`
Karate	$0.1n$	8.20	8.09	0.00	0.00	3.84
	$0.2n$	9.12	9.00	0.00	0.00	9.27
	$0.3n$	7.77	7.66	0.00	0.00	21.23
	n	6.34	6.23	0.00	0.01	191.59
Karate_R	$0.1n$	7.53	7.43	0.00	0.00	1.07
	$0.2n$	9.86	9.75	0.00	0.00	9.04
	$0.3n$	9.06	8.96	0.00	0.00	21.17
	n	6.16	6.05	0.00	0.01	190.35
Dolphine	$0.1n$	205.63	199.97	0.24	0.05	45.87
	$0.2n$	236.62	230.36	0.24	0.12	115.38
	$0.3n$	207.68	201.68	0.24	0.21	219.37
	n	150.49	144.32	0.05	0.36	2187.01
Dolphine_R	$0.1n$	194.90	188.14	0.25	0.05	47.32
	$0.2n$	233.70	227.45	0.25	0.12	114.33
	$0.3n$	233.48	227.04	0.25	0.22	218.90
	n	155.88	149.58	0.04	0.36	2185.27
Polbooks	$0.1n$	2573.13	2569.48	0.02	0.17	134.25
	$0.2n$	3227.13	3223.47	0.02	0.46	518.95
	$0.3n$	3138.83	3135.14	0.02	0.47	1106.90
	$0.4n5$	2485.89	2482.08	0.01	0.54	-
Polbooks_R	$0.1n$	3080.75	3077.09	0.02	0.16	30.30
	$0.2n$	3111.62	3107.96	0.02	0.34	865.37
	$0.3n$	2884.86	2881.20	0.05	0.37	1959.48
	$n5$	2208.06	2204.32	0.05	0.70	-
Twitter	$0.1n$	993.31	983.78	0.03	0.11	104.44
	$0.2n$	804.09	798.72	0.01	0.21	278.87
	$0.3n$	1062.83	1055.41	0.02	0.34	629.73
	n	686.92	675.53	0.01	0.38	5265.32
Twitter_R	$0.1n$	644.68	642.81	0.01	0.08	52.16
	$0.2n$	490.46	488.56	0.01	0.21	156.52
	$0.3n$	564.60	562.71	0.01	0.37	321.74
	$n0$	573.19	569.15	0.03	0.50	3454.26

7 Conclusion

In this thesis, we adopted the setting in Matakos *et al.*'s paper. Matakos *et al.* [18] define two concepts, *exposure index* and *diversity index* respectively to measure the user's exposure to different opinions and the diversity of the opinions in social network overall. Their motivation was to break the *filter bubble* in the social networks by changing the some users' *exposure index* such that the *diversity index* is maximized.

Different from their work, we cast the *exposure index* in to a continuous area, since exposures are in most cases, not extreme. We constructed two models, respectively, bounded-box diversity maximization and the ℓ_2-bounded diversity maximization. We prove that the variables of the prior model always take the extreme values to reach the optima, as a consequence, it can be transformed into the quadratic knapsack problem. While the latter one with the ℓ_2 constraint added, can no longer be transformed into a quadratic knapsack problem. To solve these models, we apply the semidefinite relaxation based methods and the greedy based methods.

We conduct experiments on five different datasets, where each one initially captures the feature of *filter bubble* and with low *diversity index*. With the experiments, we verified the effectiveness of the semidefinite relaxation based methods and the scalability of the greedy based methods.

7.1 Future Work

The solution regarding the model ℓ_2-bounded diversity maximization consumes too much time by using the semidefinite relaxation methods. It becomes infeasible even for small graphs of hundreds of nodes. The problem can be described as solving the non-convex quadratically constrained quadratic program with cardinality constraint and linear constraint. Other efficient methods can be analyzed in the future.

References

[1] News Use Across Social Media Platforms 2018 | Pew Research Center, September 2018.

[2] James Damore. Google's Ideological Echo Chamber. 2017. [Online; accessed 04-December-2018]. Retrive from: https://assets.documentcloud.org/documents/3914586/Googles-Ideological-Echo-Chamber.pdf

[3] 1615 L. St NW, Suite 800 Washington, and DC 20036 USA202-419-4300 | Main202-419-4349 | Fax202-419-4372 | Media Inquiries. Political Polarization, 1994-2017 | Pew Research Center.

[4] Michela Del Vicario, Antonio Scala, Guido Caldarelli, H Eugene Stanley, and Walter Quattrociocchi. Modeling confirmation bias and polarization. *Scientific reports*, 7:40391, 2017.

[5] Eli Pariser. *The filter bubble: What the Internet is hiding from you*. Penguin UK, 2011.

[6] E. Bakshy, S. Messing, and L. A. Adamic. Exposure to ideologically diverse news and opinion on Facebook. *Science*, 348(6239):1130–1132, June 2015.

[7] M D Conover, J Ratkiewicz, M Francisco, B Goncalves, A Flammini, and F Menczer. Political Polarization on Twitter. page 8.

[8] Lada A Adamic and Natalie Glance. The Political Blogosphere and the 2004 U.S. Election: Divided They Blog. In *Proceedings of the 3rd international workshop on Link discovery*, pages 36–43. ACM, 2005. Retrive from: http://www-personal.umich.edu/~mejn/netdata/.

[9] Alessandro Bessi, Fabiana Zollo, Michela Del Vicario, Michelangelo Puliga, Antonio Scala, Guido Caldarelli, Brian Uzzi, and Walter Quattrociocchi. Users Polarization on Facebook and Youtube. *PLOS ONE*, 11(8):e0159641, August 2016.

[10] P. Dandekar, A. Goel, and D. T. Lee. Biased assimilation, homophily, and the dynamics of polarization. *Proceedings of the National Academy of Sciences*, 110(15):5791–5796, April 2013.

[11] Pedro Henrique Calais Guerra, Wagner Meira Jr, Claire Cardie, and Robert Kleinberg. A measure of polarization on social media networks based on community boundaries. In *ICWSM*, 2013.

[12] Antonis Matakos, Evimaria Terzi, and Panayiotis Tsaparas. Measuring and moderating opinion polarization in social networks. *Data Mining and Knowledge Discovery*, 31(5):1480–1505, 2017.

[13] Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis. Quantifying controversy on social media. *ACM Transactions on Social Computing*, 1(1):3, 2018.

[14] Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis. Reducing Controversy by Connecting Opposing Views. pages 81–90. ACM Press, 2017.

[15] Cameron Musco, Christopher Musco, and Charalampos E Tsourakakis. Minimizing polarization and disagreement in social networks. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pages 369–378. International World Wide Web Conferences Steering Committee, 2018.

[16] Kiran Garimella, Aristides Gionis, Nikos Parotsidis, and Nikolaj Tatti. Balancing information exposure in social networks. In *Advances in Neural Information Processing Systems*, pages 4663–4671, 2017.

[17] Cigdem Aslay, Antonis Matakos, Esther Galbrun, and Aristides Gionis. Maximizing the Diversity of Exposure in a Social Network. *arXiv:1809.04393 [physics]*, September 2018. arXiv: 1809.04393.

[18] Antonis Matakos and Aristides Gionis. Tell me something my friends do not know: Diversity maximization in social networks. *arXiv preprint arXiv:1811.10354*, 2018.

[19] Political Polarization and Media Habits | Pew Research Center, October 2014

[20] Hans Kellerer, Ulrich Pferschy, and David Pisinger. Introduction to np-completeness of knapsack problems. In *Knapsack problems*, pages 483–493. Springer, 2004.

[21] C Helmberg, F Rendl, and R Weismantel. A semidefinite programming approach to the quadratic knapsack problem. page 19.

[22] Zhi-quan Luo, Wing-kin Ma, Anthony So, Yinyu Ye, and Shuzhong Zhang. Semidefinite relaxation of quadratic optimization problems. 27(3):20–34.

[23] Morris H DeGroot. Reaching a consensus. *Journal of the American Statistical Association*, 69(345):118–121, 1974.

[24] Noah E Friedkin and Eugene C Johnsen. Social positions in influence networks. *Social Networks*, 19(3):209–222, 1997.

[25] Waugh, Andrew Scott and Pei, Liuyi and Fowler, James H and Mucha, Peter J and Porter, Mason Alexander Party polarization in congress: A network science approach, 2009

[26] Akoglu, Leman. Quantifying political polarity based on bipartite opinion networks. *Eighth International AAAI Conference on Weblogs and Social Media*. 2014.

[27] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1):265–294, December 1978.

[28] Stephen P. Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, Cambridge, UK ; New York, 2004.

[29] Lieven Vandenberghe and Stephen Boyd. Semidefinite programming. *SIAM review*, 38(1):49–95, 1996.

[30] Farid Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM journal on Optimization*, 5(1):13–51, 1995.

[31] László Lovász and Alexander Schrijver. Cones of matrices and set-functions and 0–1 optimization. *SIAM journal on optimization*, 1(2):166–190, 1991.

[32] Michel X Goemans and David P Williamson. Improved Approximation Algorithms for Maximum Cut and Satis ability Problems Using Semidefinite Programming. page 27.

[33] M. S. Bazaraa, Hanif D. Sherali, and C. M. Shetty. *Nonlinear programming: theory and algorithms*. Wiley-Interscience, Hoboken, N.J, 3rd ed edition, 2006. OCLC: ocm61478842.

[34] Roger A Horn, Roger A Horn, and Charles R Johnson. *Matrix analysis*. Cambridge university press, 1990.

[35] Alexandre D'aspremont, Laurent E Ghaoui, Michael I Jordan, and Gert R Lanckriet. A direct formulation for sparse PCA using semidefinite programming. page 8.

[36] Yinyu Ye. Approximating quadratic programming with bound and quadratic constraints. *Mathematical Programming*, 84(2):219–226, February 1999.

[37] Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

[38] David J. Rader Jr. and Gerhard J. Woeginger. The quadratic 0-1 knapsack problem with series-parallel support. *Oper. Res. Lett.*, 30(3):159–166, June 2002.

[39] David Pisinger. The quadratic knapsack problem—a survey. *Discrete Applied Mathematics*, 155(5):623 – 648, 2007.

[40] G. Gallo, P. L. Hammer, and B. Simeone. *Quadratic knapsack problems*, pages 132–149. Springer Berlin Heidelberg, Berlin, Heidelberg, 1980.

[41] Alain Billionnet and Frédéric Calmels. Linear programming for the 0–1 quadratic knapsack problem. *European Journal of Operational Research*, 92:310–325, 07 1996.

[42] Alberto Caprara, David Pisinger, and Paolo Toth. Exact solution of the quadratic knapsack problem. *INFORMS J. on Computing*, 11(2):125–137, February 1999.

[43] Paul Chaillou, Pierre Hansen, and Yvon Mahieu. Best network flow bounds for the quadratic knapsack problem. In Bruno Simeone, editor, *Combinatorial Optimization*, pages 225–235, Berlin, Heidelberg, 1989. Springer Berlin Heidelberg.

[44] Philippe Michelon and Louis Veilleux. Lagrangean methods for the 0–1 quadratic knapsack problem. *European Journal of Operational Research*, 92(2):326–341, 1996.

[45] Alain Billionnet, Alain Faye, and Éric Soutif. A new upper bound for the 0-1 quadratic knapsack problem. *European journal of operational research*, 112(3):664–672, 1999.

[46] Bernard Chazelle, Carl Kingsford, and Mona Singh. A semidefinite programming approach to side chain positioning with new rounding strategies. *INFORMS Journal on Computing*, 16(4):380–392, 2004.

[47] Bryant A Julstrom. Greedy, genetic, and greedy genetic algorithms for the quadratic knapsack problem. In *Proceedings of the 7th annual conference on Genetic and evolutionary computation*, pages 607–614. ACM, 2005.

[48] Wayne W. Zachary. An Information Flow Model for Conflict and Fission in Small Groups, 1977. Retrive from: https://sites.google.com/site/ucinetsoftware/datasets/zacharykarateclub.

[49] David Lusseau, Karsten Schneider, Oliver J Boisseau, Patti Haase, Elisabeth Slooten, and Steve M Dawson. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations, 2003. Retrive from: http://www-personal.umich.edu/~mejn/netdata/.

[50] Valdis Krebs. Books about us politics data set [data set], 2004. Retrive from: http://www-personal.umich.edu/~mejn/netdata/.

[51] Jure Leskovec and Christos Faloutsos. Sampling from large graphs. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '06*, page 631, Philadelphia, PA, USA, 2006. ACM Press.

A More results of ℓ_2-bounded diversity maximization

Table A1: Result of ℓ_2-bounded diversity maximization

Dataset(Diversity Index)	k	α	$\|\|\mathbf{x}\|\|_2^2$	GaussianSDPtop	SDPMax	$\|\|\mathbf{x}\|\|_2^2$	TwoStepSDP	$\|\|\mathbf{x}\|\|_2^2$	AbGreedy	$\|\|\mathbf{x}\|\|_2^2$	RtGreedy
Karate (12.64)	$0.1n$	7.64	5.19	72.91	145.02	7.16	91.83	5.11	91.83	7.16	91.83
	$0.2n$	7.64	4.66	76.61	148.51	7.60	86.64	7.16	105.57	7.16	91.83
	$0.3n$	7.64	6.74	87.24	149.44	7.59	101.94	7.16	105.57	7.16	91.83
	n	7.64	7.43	104.38	149.54	7.37	95.27	7.16	105.57	7.16	91.83
Karate_R (44.23)	$0.1n$	7.64	1.45	84.05	196.94	1.45	84.05	1.89	82.42	2.11	84.69
	$0.2n$	7.64	1.90	104.52	206.52	6.83	100.19	5.16	102.45	6.46	115.41
	$0.3n$	7.64	2.88	117.75	210.08	7.58	119.72	5.85	108.82	6.46	115.41
	n	7.64	3.10	135.51	210.39	7.23	137.15	5.85	108.82	6.46	115.41
Dolphine (21.96)	$0.1n$	14.45	7.79	109.07	218.42	14.31	118.89	12.39	136.26	12.73	144.78
	$0.2n$	14.45	12.55	149.32	226.68	13.99	160.97	12.39	136.26	12.73	144.78
	$0.3n$	14.45	13.46	158.82	228.25	14.00	157.66	12.39	136.26	12.73	144.78
	n	14.45	14.33	180.89	228.25	14.42	188.97	12.39	136.26	12.73	144.78
Dolphine_R (90.44)	$0.1n$	14.45	3.16	147.03	327.62	7.84	159.33	10.35	170.17	12.19	175.19
	$0.2n$	14.45	10.58	189.00	341.12	13.85	201.30	14.31	183.65	14.28	205.60
	$0.3n$	14.45	12.51	212.84	346.10	13.71	204.94	14.31	183.65	14.28	205.60
	n	14.45	12.55	244.50	346.34	13.69	278.73	14.31	183.65	14.28	205.60
Twitter (247.72)	$0.1n$	18.50	8.79	780.94	1501.35	15.37	962.31	15.61	992.89	15.74	1000.61
	$0.2n$	18.50	12.64	995.10	1555.90	18.47	1115.65	17.90	1063.77	15.74	1000.61
	$0.3n$	18.50	14.22	1068.03	1579.43	18.33	1282.65	17.90	1063.77	15.74	1000.61
	n	18.50	16.74	1337.55	1586.04	18.50	1567.78	17.90	1063.77	15.74	1000.61
Twitter_R (842.03)	$0.1n$	18.50	4.92	1201.46	2378.80	10.82	1292.15	12.43	1292.25	8.96	1302.78
	$0.2n$	18.50	10.07	1478.08	2514.37	18.36	1336.40	18.34	1413.25	17.77	1551.10
	$0.3n$	18.50	12.93	1722.16	2589.14	18.25	1593.84	18.34	1413.25	17.77	1551.10
	n	18.50	12.69	2300.02	2637.10	18.49	2561.22	18.34	1413.25	17.77	1551.10
Polbooks (98.30)	$0.1n$	24.08	-	-	-	24.04	440.67	23.29	467.08	23.48	448.83
	$0.2n$	24.08	-	-	-	23.45	461.12	23.29	467.08	23.48	448.83
	$0.3n$	24.08	-	-	-	23.95	478.52	23.29	467.08	23.48	448.83
	n	24.08	-	-	-	-	-	23.29	467.08	23.48	448.83
Polbooks_R (254.57)	$0.1n$	24.08	-	-	-	10.04	439.90	12.77	463.10	14.97	471.94
	$0.2n$	24.08	-	-	-	23.55	544.56	23.69	557.94	22.43	565.44
	$0.3n$	24.08	-	-	-	22.71	533.19	23.69	557.94	22.43	565.44
	$n5$	24.08	-	-	-	-	-	23.69	557.94	22.43	565.44

Table A2: Running Time(s) of ℓ_2-bounded diversity maximization

Dataset	k	GaussianSDPtop	SDPMax	TwoStepSDP	AbGreedy	RtGreedy
Karate	$0.1n$	5.87	5.80	0.64	0.00	0.00
	$0.2n$	4.91	4.84	5.03	0.00	0.00
	$0.3n$	5.63	5.56	11.34	0.00	0.00
	n	3.64	3.57	104.44	0.00	0.01
Karate_R	$0.1n$	5.65	5.59	0.62	0.00	0.00
	$0.2n$	4.89	4.82	5.05	0.00	0.00
	$0.3n$	5.15	5.08	11.22	0.00	0.00
	n	3.91	3.84	106.19	0.00	0.01
Dolphine	$0.1n$	126.45	120.99	30.30	0.34	0.05
	$0.2n$	143.41	137.87	69.52	0.34	0.13
	$0.3n$	126.50	120.85	129.08	0.34	0.23
	n	76.56	70.81	1204.72	0.04	0.46
Dolphine_R	$0.1n$	126.52	121.09	15.90	0.06	0.05
	$0.2n$	126.49	121.01	66.89	0.34	0.13
	$0.3n$	154.47	148.84	127.13	0.34	0.22
	n	76.39	70.84	1213.74	0.04	0.46
Twitter	$0.1n$	542.25	534.83	20.25	0.02	0.10
	$0.2n$	587.79	580.05	182.46	0.02	0.25
	$0.3n$	588.52	580.73	371.78	0.02	0.45
	n	403.80	395.93	3602.38	0.02	0.49
Twitter_R	$0.1n$	429.17	423.70	16.15	0.01	0.08
	$0.2n$	469.25	463.74	151.61	0.02	0.22
	$0.3n$	509.30	503.71	338.15	0.02	0.38
	n	388.57	382.94	3230.94	0.02	0.48
Polbooks	$0.1n$	-	-	139.64	0.03	0.18
	$0.2n$	-	-	481.54	0.03	0.45
	$0.3n$	-	-	1122.47	0.03	0.47
	n	-	-	-	0.02	0.54
Polbooks_R	$0.1n$	-	-	27.10	0.03	0.18
	$0.2n$	-	-	491.78	0.05	0.45
	$0.3n$	-	-	1113.65	0.05	0.46
	n	-	-	-	0.05	0.56

,

Table A3: Result of ℓ_2-bounded diversity maximization

Dataset(Diversity Index)	k	α	$\|\|\mathbf{x}\|\|_2^2$	`GaussianSDPtop`	`SDPMax`	$\|\|\mathbf{x}\|\|_2^2$	`TwoStepSDP`	$\|\|\mathbf{x}\|\|_2^2$	`AbGreedy`	$\|\|\mathbf{x}\|\|_2^2$	`RtGreedy`
Karate (12.64)	$0.1n$	15.27	6.46	71.95	217.46	7.16	91.83	5.11	91.83	7.16	91.83
	$0.2n$	15.27	9.87	97.26	233.38	10.63	121.84	10.32	116.79	12.07	123.54
	$0.3n$	15.27	8.35	113.50	241.03	15.24	120.29	12.72	117.29	14.87	149.00
	n	15.27	15.20	159.58	243.23	13.45	149.73	12.72	117.29	15.19	154.67
Karate_R (44.23)	$0.1n$	15.27	1.41	84.69	271.17	1.45	84.05	1.89	82.42	2.11	84.69
	$0.2n$	15.27	2.65	109.08	290.25	10.07	100.42	5.16	102.45	6.46	115.41
	$0.3n$	15.27	6.22	123.40	298.98	14.94	127.46	11.32	117.99	12.09	142.46
	n	15.27	14.43	165.50	301.08	15.04	147.36	15.15	137.51	13.53	154.23
Dolphine (21.96)	$0.1n$	28.89	12.38	123.89	382.16	15.56	126.20	12.39	136.26	12.73	144.78
	$0.2n$	28.89	16.35	172.96	397.74	27.12	197.01	24.27	201.87	24.24	212.13
	$0.3n$	28.89	20.94	209.76	402.46	28.80	199.07	28.85	219.61	27.65	241.18
	n	28.89	23.50	263.31	402.76	26.99	269.23	28.85	219.61	27.65	241.18
Dolphine_R (90.44)	$0.1n$	28.89	10.65	158.52	490.11	7.84	159.33	10.35	170.17	12.19	175.19
	$0.2n$	28.89	16.23	209.66	511.55	18.53	208.36	22.52	212.31	16.91	234.67
	$0.3n$	15.27	8.35	113.50	241.03	15.24	120.29	12.72	117.29	14.87	149.00
	n	15.27	15.20	159.58	243.23	13.45	149.73	12.72	117.29	15.19	154.67
Twitter (247.72)	$0.1n$	36.99	13.07	957.31	2477.26	15.37	962.31	15.61	992.89	15.74	1000.61
	$0.2n$	36.99	20.04	1190.88	2542.59	35.53	1367.01	33.87	1399.95	31.42	1472.40
	$0.3n$	36.99	23.00	1386.36	2561.44	34.42	1599.63	36.34	1428.61	36.51	1613.34
	n	36.99	27.83	1769.08	2562.05	36.98	2370.80	36.34	1428.61	36.51	1613.34
Twitter_R (842.03)	$0.1n$	36.99	8.80	1202.98	3428.07	10.82	1292.15	12.43	1292.25	8.96	1302.78
	$0.2n$	36.99	12.73	1492.82	3558.87	19.31	1528.83	23.55	1500.61	19.96	1629.49
	$0.3n$	36.99	8.99	1718.55	3590.86	32.77	1782.34	33.00	1607.37	27.22	1917.77
	n	36.99	13.30	2331.98	3594.31	29.25	2355.36	35.68	1665.12	36.10	2281.51
Polbooks (98.30)	$0.1n$	48.15	-	-	-	25.64	461.38	23.29	467.08	25.88	469.53
	$0.2n$	48.15	-	-	-	45.03	554.72	47.96	553.78	43.31	637.00
	$0.3n$	48.15	-	-	-	47.69	574.05	47.96	553.78	47.44	674.30
	n	48.15		-	-	-	-	47.96	553.78	47.44	674.30
Polbooks_R (254.57)	$0.1n$	48.15	-	-	-	10.04	439.90	12.77	463.10	14.97	471.94
	$0.2n$	48.15	-	-	-	34.90	543.84	26.90	592.10	27.20	632.56
	$0.3n$	48.15	-	-	-	34.78	625.01	40.66	663.16	38.75	738.44
	n	48.15	-	-	-	-		47.28	734.88	48.08	796.32

Table A4: Running Time(s) of ℓ_2-bounded diversity maximization

Dataset	k	GaussianSDPtop	SDPMax	TwoStepSDP	AbGreedy	RtGreedy
Karate	$0.1n$	6.23	6.16	0.61	0.00	0.00
	$0.2n$	5.01	4.94	0.61	0.00	0.00
	$0.3n$	4.73	4.66	11.49	0.00	0.00
	n	3.49	3.42	105.78	0.00	0.01
Karate_R	$0.1n$	5.73	5.67	0.60	0.00	0.00
	$0.2n$	4.24	4.17	0.62	0.00	0.00
	$0.3n$	4.50	4.43	11.59	0.00	0.00
	n	3.76	3.69	104.79	0.00	0.01
Dolphine	$0.1n$	129.91	124.33	13.87	0.03	0.05
	$0.2n$	153.80	148.16	13.86	0.04	0.13
	$0.3n$	188.55	182.77	125.81	0.36	0.22
	n	84.39	78.45	1223.79	0.06	0.46
Dolphine_R	$0.1n$	133.14	127.41	14.05	0.11	0.05
	$0.2n$	149.95	144.35	14.43	0.08	0.14
	$0.3n$	211.95	206.33	130.86	0.37	0.23
	n	101.43	95.71	1223.22	0.09	0.46
Twitter	$0.1n$	692.52	684.03	18.04	0.04	0.10
	$0.2n$	647.85	639.39	20.08	0.05	0.27
	$0.3n$	820.87	812.21	392.24	0.05	0.45
	n	407.94	399.57	3841.08	0.04	0.49
Twitter_R	$0.1n$	545.14	537.43	16.63	0.03	0.10
	$0.2n$	624.22	616.37	17.04	0.04	0.27
	$0.3n$	671.50	663.63	21.51	0.07	0.47
	n	403.90	398.24	3281.85	0.06	0.48
Polbooks	$0.1n$	-	-	23.52	0.04	0.18
	$0.2n$	-	-	24.16	0.06	0.45
	$0.3n$	-	-	1155.42	0.06	0.47
	n	-	-	-	0.06	0.56
Polbooks_R	$0.1n$	-	-	24.39	0.03	0.18
	$0.2n$	-	-	26.16	0.07	0.45
	$0.3n$	-	-	27.52	0.09	0.47
	n	-	-	-	0.14	0.56